

**MOBILE APP ONLINE MENGGUNAKAN LOCATION BASED
SERVICE & METODE HAVERSINE DALAM
PENGIMPLEMENTASIAN
JARAK & MEDAN**

SKRIPSI

**Oleh :
MUHAMMAD MUSLIM
14650008**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

**MOBILE APP ONLINE MENGGUNAKAN LOCATION BASED SERVICE &
METODE HAVERSINE DALAM
PENGIMPLEMENTASIAN
JARAK & MEDAN**

SKRIPSI

Diajukan kepada:

**Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh :

**Muhammad Muslim
NIM.14650008**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

LEMBAR PERSETUJUAN

**MOBILE APP ONLINE MENGGUNAKAN LOCATION BASED SERVICE &
METODE HAVERSINE DALAM
PENGIMPLEMENTASIAN
JARAK & MEDAN**

SKRIPSI

Oleh :

Muhammad Muslim

NIM. 14650008

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 2018

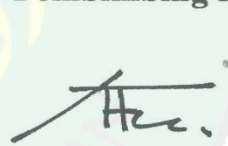
Pembimbing I



Dr. M. Amin Hariyadi, M.T

NIP. 19670118 200501 1 001

Pembimbing II



Fatchurrochman, M.Kom

NIP. 19700731 200501 1 002

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdian

NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

**MOBILE APP ONLINE MENGGUNAKAN LOCATION BASED SERVICE &
METODE HAVERSINE DALAM
PENGIMPLEMENTASIAN
JARAK & MEDAN**

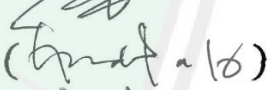
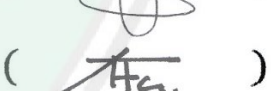
SKRIPSI

Oleh :

Muhammad Muslim
NIM. 14650008

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal : 6 Maret 2018

Penguji Utama	:	<u>Dr. Cahyo Crysdian</u>	()
		NIP. 19740424 200901 1 008	
Ketua Penguji	:	<u>Linda Salma Angreani, M.T</u>	()
		NIP. 19770803 200912 2 005	
Sekretaris Penguji	:	<u>Dr. M. Amin Hariyadi, M.T</u>	()
		NIP. 19670118 200501 1 001	
Anggota Penguji	:	<u>Fatchurrochman, M.Kom</u>	()
		NIP. 19700731 200501 1 002	

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Muslim

NIM : 14650008

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : **MOBILE APP ONLINE MENGGUNAKAN LOCATION BASED SERVICE & METODE HAVERSINE DALAM PENGIMPLEMENTASIAN JARAK & MEDAN**

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-nenar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 Maret 2018

Yang membuat pernyataan



Muhammad Muslim

NIM. 14650008

Motto

Dahulukanlah Allah....

Maka Allah akan mendahulukan mu atas segala urusan mu



Halaman Persembahan

Ananda muslim persembahkan untuk ibu & ayah yang telah memberi kesempatan & kepercayaan kepada ananda untuk menimba ilmu ditempat yang cukup jauh dari tempat keluarga tinggal. Kepada ibu, ananda tidak akan pernah mengkhianati tetesan air mata ibu yang membesarkan ananda dalam dekapan do'a kasih sayang. Begitu pula ayah, ananda tidak akan pernah mengkhianati tetesan keringat ayah yang mencukupi kebutuhan ananda dalam peluhnya dunia. jauh ananda menimba ilmu berharap pulang membawakan jannah kepada ibu & ayah melalui ilmu yang bermanfaat yang telah ananda dapat.

KATA PENGANTAR



Assalamualaikum warahmatullahi wabarakaatuh.

Segala puji bagi Allah SWT Tuhan semesta alam, yang telah melimpahkan rahmat serta karunia-Nya kepada penulis sehingga dapat menyelesaikan skripsi dengan judul “Pencarian Rute Transportasi Umum Kota Malang dengan Algoritma *Best-Path Planning* Berbasis Android” dengan baik dan lancar.

Shalawat serta salam semoga tercurahkan kepada Nabi Agung Muhammad SAW, insan mulia yang telah menghabiskan waktu untuk menuntun umatnya dari gelapnya kekufuran kearah keselamatan hidup, menuju cahaya Islam yang terang benderang, Islam yang *rahmatan lil ‘aalamin*.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, mungkin akan sulit bagi penulis untuk menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati, penulis ucapkan terimakasih kepada :

1. Bapak Dr. M. Amin Hariyadi, M.T selaku dosen pembimbing I serta sebagai dosen wali yang telah membimbing, memberi masukan dan saran ketika penulis mengalami kesulitan selama proses perkuliahan dari semester awal hingga semester akhir, kemudahan serta memberi kepercayaan kepada penulis dalam pengerjaan skripsi.
2. Bapak Fatchurrochman, M.Kom selaku dosen pembimbing II yang selalu memberikan dukungan, nasehat serta masukan yang memabngun dalam penyusunan skripsi.

3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang sudah memberi banyak pengetahuan, inspirasi dan pengalaman yang berharga.
4. Seluruh dosen dan staff Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan bimbingan, arahan, mengalirkan ilmu pengetahuan, pengalaman serta wawasan sebagai pedoman dan bekal bagi penulis.
5. Ibu dan ayah tercinta, serta keluarga besar yang selalu memberi dukungan yang tak terhingga serta doa yang senantiasa mengiringi setiap langkah penulis serta menjadi motivasi luar biasa kepada penulis dalam menuntut ilmu dan menyelesaikan karya ini.
7. Teman-teman Jurusan Teknik Informatika angkatan 2014 (Biner), teman-teman komunitas Fun Java serta teman-teman kampus tercinta di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan masukan yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peniliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. Aamiin ya rabbal ‘aalamin.

Wassalamualaikum warahmatullahi wabarakaatuh.

Malang, 20 Maret 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
Motto	v
LEMBAR PERSEBAHAN	vi
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
ABSTRAK	xiii
ABSTRACT	xiv
ملخص	xv
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Tujuan Penelitian	4
1.4. Batasan Penelitian	5
1.5. Manfaat Penelitian	5
BAB II	6
STUDI LITERATUR	6
2.1. <i>Location Based Service</i>	6
2.1.1. <i>Global Positioning System</i>	6
2.2. <i>Haversine</i>	9
2.3. <i>One-way Anova</i>	11
2.4. Penelitian Terkait	12
BAB III	16
METODOLOGI PENELITIAN	17
3.1. Kondisi Lingkungan Penelitian	17
3.2. Perancangan <i>System</i>	18
3.2.1. Akuisisi Data	18
3.2.2. Perancangan Proses	19
3.2.2.1 Desain <i>System</i>	20

3.2.3. Perancangan Aplikasi	36
3.3. Implementasi <i>System</i>	52
3.4. Pengujian <i>System</i>	53
3.5. Penarikan Kesimpulan Sementara	55
BAB IV	56
IMPLEMENTASI DAN PENGUJIAN SISTEM	56
4.1. Implementasi <i>Interface</i>	57
4.1.1. Tampilan <i>Splash Screen</i>	57
4.1.2. <i>Menu Utama</i>	59
4.1.3. <i>Menu Rute Maps</i>	60
4.1.4. <i>Menu History</i>	62
4.1.5. <i>Menu Deposit</i>	64
4.1.6. <i>Menu Driver</i>	65
4.1.7. <i>Menu Monitoring Penumpang</i>	66
4.2. Pengujian Sistem.....	68
4.2.1. Persiapan Data	69
4.2.2. Pengujian	70
4.2.3. Hasil Analisa	75
4.3. Integrasi Islam.....	77
BAB V	82
PENUTUP	82
5.1. Kesimpulan	82
5.2. Saran	83
DAFTAR PUSTAKA.....	84

DAFTAR GAMBAR

Gambar 2.1 Area pertama, kedua dan ketiga.....	8
Gambar 2.2 <i>Triangulation</i>	8
Gambar 2.3 Segitiga bola diselesaikan dengan hukum <i>Havarsine Formula</i>	10
Gambar 2.4 <i>Random Generated Map</i>	14
Gambar 3.1 Alur metode penelitian	16
Gambar 3.2 <i>Blok Diagram</i> akuisisi data.....	19
Gambar 3.3 Desain sistem.....	20
Gambar 3.4 Arsitektur kerja LBS.....	22
Gambar 3.5 Komponen LBS	23
Gambar 3.6 <i>Euclidean 2 dimensi</i>	26
Gambar 3.7 Rumus busur	28
Gambar 3.8 Kontur kemiringan pada <i>google earth</i>	30
Gambar 3.9 <i>Output</i> aplikasi.....	35
Gambar 3.10 <i>Use Case Diagram</i>	40
Gambar 3.11 <i>Activity Diagram</i> <i>signup member</i> atau ojek	41
Gambar 3.12 <i>Activity Diagram</i> <i>login</i>	42
Gambar 3.13 <i>Activity Diagram</i> cari tempat/lokasi tujuan	43
Gambar 3.14 <i>Activity Diagram</i> pilih ojek	44
Gambar 3.15 <i>Activity Diagram</i> melihat penumpang.....	45
Gambar 3.16 <i>Activity Diagram</i> terima order	46
Gambar 3.17 <i>Activity Diagram</i> tentang aplikasi	47
Gambar 3.18 <i>Sequance Diagram</i> ojek online.....	48
Gambar 3.19 <i>Class Diagram</i> ojek online	49
Gambar 3.20 <i>Menu login</i>	51
Gambar 3.21 <i>Menu utama</i>	51
Gambar 3.22 <i>Menu</i> pencarian tempat atau lokasi.....	52
Gambar 4.1 Tampilan <i>splash screen</i>	58
Gambar 4.2 <i>Source code</i> <i>activity splash screen</i>	58
Gambar 4.3 <i>Menu utama</i>	60
Gambar 4.4 <i>Source code</i> <i>activity main</i>	60
Gambar 4.5 <i>Menu rute maps</i>	61
Gambar 4.6 <i>Source code</i> <i>activity maps</i>	62
Gambar 4.7 <i>Menu history</i>	63
Gambar 4.8 <i>Source code</i> <i>activity history</i>	63
Gambar 4.9 <i>Menu deposit</i>	64
Gambar 4.10 <i>Source code</i> <i>activity deposit</i>	65
Gambar 4.11 Tampilan <i>menu drive</i>	66
Gambar 4.12 <i>Source code</i> <i>activity main driver</i>	66
Gambar 4.13 <i>Menu</i> pencarian penumpang.....	67
Gambar 4.14 <i>Source code</i> <i>activity main driver</i>	68
Gambar 4.15 Grafik perbandingan jarak	74
Gambar 4.16 Grafik perbandingan tarif harga.....	75

DAFTAR TABEL

Tabel 3.1 Letak daerah, jarak wilayah tiap kabupaten, luas wilayah dan ketinggian wilayah	17
Tabel 3.2 Perhitungan <i>Euclid</i> dan <i>Haversine</i>	29
Tabel 3.3 Perubahan tarif harga.....	34
Tabel 3.4 Batas tarif bawah & atas.....	35
Tabel 3.5 <i>Scenario Diagram</i> menu daftar Sebagai Member/Ojek	36
Tabel 3.6 <i>Scenario Diagram</i> menu login.....	36
Tabel 3.7 <i>Scenario Diagram</i> menu cari tempat/lokasi	37
Tabel 3.8 <i>Scenario Diagram</i> menu pilih ojek.....	38
Tabel 3.9 <i>Scenario Diagram</i> menu lihat penumpang.....	38
Tabel 3.10 <i>Scenario Diagram</i> menu terima order.....	39
Tabel 3.11 <i>Skenario Diagram</i> menu tentang aplikasi.....	39
Tabel 3.12 <i>Testing</i> metode.....	54
Tabel 4.1 Data <i>testing</i>	66
Tabel 4.2 Proses <i>testing</i>	68
Tabel 4.3 Hasil <i>testing</i>	68

ABSTRAK

Muslim, Muhammad. 2018. ***Mobile App Online Menggunakan Location Based Service & Metode Haversine Dalam Pengimplementasian Jarak & Medan***. Skripsi. Teknik Informatika Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Dr. M. Amin Hariyadi, M.T, (II) Fatchurrochman, M.Kom

Kata Kunci : *Mobile App Online, Location Based Service, Haversine*, Implementasi jarak dan medan

Jarak merupakan angka yang menunjukkan seberapa jauh suatu benda berubah posisi melalui suatu lintasan tertentu. Dalam pengertian lain, jarak dapat berupa estimasi jarak fisik dari dua buah posisi berdasarkan kriteria tertentu. Dan medan merupakan penggambaran karakter area atau wilayah yang dapat berupa kontur kemiringan atau ketinggian daerah.

Pada penelitian ini mengangkat sisi integrasi penggunaan transportasi jarak dekat dan teknologi *mobile* yang terotomasi sehingga memudahkan dalam memperkirakan estimasi jarak, medan yang dilalui dan tarif. Dari penelitian yang telah dilakukan dengan menggunakan *Location Based Service* dan metode *Haversine* dalam pengimplementasian jarak dan medan didapatkan hasil berupa proses penggambaran rute berdasarkan jarak yang telah dihitung dengan persentase akurasi dari metode 98% untuk jarak dan 95% untuk medan. Sementara hasil persentase dari perbandingan akurasi aplikasi yang telah dikembangkan dengan aplikasi yang telah ada didapatkan selisih jarak 0.3% serta 0.24% untuk perbandingan estimasi harga, sehingga dapat disimpulkan bahwa aplikasi yang telah dikembangkan lebih efektif dan efisien dalam estimasi jarak, medan, dan tarif.

ABSTRACT

Muslim, Muhammad. 2018. **Mobile App Online Using Location Based Service & Method Haversine In Implementation Distance & Terrain.** Undergraduate Thesis. Department Of Engineering Informatics, Faculty Science And Technology. State Islamic University Maulana Malik Ibrahim Malang. Advisers : (I) Dr. M. Amin Hariyadi, M.T, (II) Fatchurrochman, M.Kom

Keyword : Mobile App Online, Location Based Service, Haversine, Implementation Distance & Terrain.

The distance is the number that shows how far an object changed position through a particular path. In another sense, the distance can be estimated physical distance from two positions based on certain criteria. And the terrain is the depiction of the character of the area or region can be either the height or the slope of the contours of the area.

On the research side raised the use of close-range transport integration and mobile technologies are making it easier for automated estimation of the distance estimates, in terrain that has endured, and price. From the research that has been done using lbs and methods in the implementation of haversine distance and terrain obtained results in the form of process route depiction based on the distance that has been calculated by the percentage accuracy of 98% for method 95% for distance and terrain. While the results of the comparison of the percentage accuracy of applications that have been developed with existing applications obtained difference in distance of 0.3% and 0.24% in comparison to the estimated price, so it can be concluded that the application has been developed more effective and efficient in the estimation of distance, terrain, and price.

ملخص

مسلم، محمد. ٨١٠٢. موبايل التطبيق علي الإنترنت باستخدام الموقع القائم علي الخدمات والأساليب في تنفيذ المسافة والتضاريس. أطروحة. كلية العلوم والتكنولوجيا. المعلوماتية هندسه . الجماعة الإسلامية الهوكوميه مولانا مالك إبراهيم مالانغ. المشرف : (١) الدكتور م. أمين هارياي ، م. ت، (٢) فاتورحم ، م. كوم

الكلمة : موبايل التطبيق علي الإنترنت, الخدمة القائمة علي الموقع, هافيرسين, تنفيذ المسافات والتضاريس.

المسافة هي الرقم الذي يظهر مدي تغير الكائن موضع من خلال مسار معين. وبمعني آخر ، فان المسافة يمكن ان تكون مقدره علي المسافة المادية من وظيفتين استنادا إلى معايير معينه. والتضاريس هو تصوير للشخصية المنطقة أو المنطقة يمكن ان يكون اما ارتفاع أو ميل من ملامح المنطقة.

في تطبيق تطوير استخدام الإنترنت باعتبارها قاعده بيانات الوصول إلى التنمية التكنولوجية بحيث يمكن الوصول اليها بغض النظر عن موقف المستخدم طالما ان هناك اتصال بالإنترنت بسبب تنفيذ المتقدمة ، وخدمات الويب ، وسحابه. وكذلك تنفيذ الحوالات كوسيلة لقياس مسافة المكان الذي يتم إرساله ، والخدمة القائمة علي الموقع كمحدد لموقع المستخدم. وفي الجانب البحثي ، زاد استخدام تكامل النقل القريب والتكنولوجيات المتنقلة مما يسهل التقدير الألى لتقديرات المسافات ، في التضاريس التي يتم اجتيازها والمعدل.

من البحوث التي تم القيام به باستخدام الرطل والأساليب في تنفيذ المسافات والتضاريس الحصول علي نتائج في شكل عمليه تصوير المسار استنادا إلى المسافة التي تم حسابها من قبل نسبه مئوية من دقه ٩٨٪ لطريقه ٩٥٪ في المائة للمسافة والتضاريس. وفي حين ان نتائج المقارنة بين دقه النسبة المئوية للتطبيقات التي تم تطويرها باستخدام التطبيقات الحالية قد حصلت علي فرق في المسافة بين ٠,٣٪ في المائة و ٠,٢٤٪ في المائة مقارنة بالسعر المقدر ، فانه يمكن الخلوصل إلى ان الطلب قد وضعت أكثر فعاليه وكفاءه في تقدير المسافة ، والتضاريس ، والسعر.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan banyak cara dan bentuk dalam berhubungan dengan sesama manusia atau berniaga sehingga bila tidak tertanam konsep keadilan dalam melakukannya, maka aktifitas berhubungan dengan sesama atau perniagaan tersebut akan ada banyak kecurangan didalamnya, baik mengurangi takaran atau haknya. Salah satu ayat yang terdapat dalam Al - Qur'an yaitu pada surat An-Nahl ayat 90, adapun ayat tertulis seperti dibawah ini.

إِنَّ اللَّهَ يَأْمُرُ بِالْعَدْلِ وَالْإِحْسَانِ وَإِيتَايَ ذِي الْقُرْبَىٰ وَيَنْهَىٰ عَنِ الْفَحْشَاءِ وَالْمُنْكَرِ وَالْبَغْيِ
يُعْظِمُ لَكُمْ لَعْنَكُمْ تَذَكَّرُونَ ﴿٩٠﴾

“Sesungguhnya Allah menyuruh (kamu) berlaku adil dan berbuat kebajikan, memberi kepada kaum kerabat, dan Allah melarang dari perbuatan keji, kemungkaran dan permusuhan. Dia memberi pengajaran kepadamu agar kamu dapat mengambil pelajaran.” (QS. An-Nahl:90).

Diterangkan pada kitab tafsir Al-Jami' liahkam al-Qur'an wa al-Mubayyin Lima Tadammanahu Min as-Sunnah wa Ayi al-Furqan yang ditafsirkan Allah memerintahkan para hamba-Nya untuk berlaku adil dalam setiap perkataan dan perbuatan. Allah menyuruh mereka untuk selalu berusaha menuju yang lebih baik dalam setiap usaha dan mengutamakan yang terbaik dari lainnya. Allah memerintahkan mereka untuk memberikan apa yang dibutuhkan oleh para kerabat sebagai cara untuk memperkokoh ikatan kasih sayang antar keluarga. Allah melarang mereka berbuat dosa, lebih-lebih dosa yang amat

buruk dan segala perbuatan yang tidak dibenarkan oleh syariat dan akal sehat. Allah melarang mereka menyakiti orang lain. Dengan perintah dan larangan itu, Allah bermaksud membimbing kalian menuju kemaslahatan dalam setiap aspek kehidupan, agar kalian selalu ingat karunia-Nya dan menaati firman-firman-Nya (Al-Qurtubi ,620 H).

Begitu juga pada perkembangan berniaga atau usaha dengan memanfaatkan kendaraan tidak hanya terbatas pada yang penggunaan binatang sebagai alat transportasi untuk berpergian melainkan telah berkembang menggunakan terapan modernisasi yang mendukung penggunaan ilmu terapan yang digabungkan dengan kemajuan teknologi yang ada, maka akan melahirkan kendaraan *modern* yang lebih efektif dan efisien dalam penggunaannya. Jika melihat perkembangan yang terjadi pada saat ini penggunaan kendaraan roda dua lebih diminati oleh masyarakat menengah kebawah, yang menggunakan kendaraan roda dua bermotor yang dikolaborasikan dengan penggunaan teknologi digital sebagai fitur yang memudahkan dalam penggunaannya. Dalam Al-Qur'an telah dicontohkan cara berniaga yang sesuai dengan konsep yang ada dalam islam, seperti yang disebutkan pada Al-Qur'an surah Al-Baqarah ayat 198.

لَيْسَ عَلَيْكُمْ جُنَاحٌ أَنْ تَبْتَغُوا فَضْلًا مِنْ رَبِّكُمْ فَإِذَا أَفَضْتُمْ مِنْ عَرَفَاتٍ

فَاذْكُرُوا اللَّهَ عِنْدَ الْمَشْعَرِ الْحَرَامِ وَاذْكُرُوهُ كَمَا هَدَانَكُمْ

وَإِنْ كُنْتُمْ مِنْ قَبْلِهِ لَمَنِ الضَّالِّينَ ﴿١٩٨﴾

“Tidak ada dosa bagimu untuk mencari karunia (rezeki hasil perniagaan) dari Tuhanmu. Maka apabila kamu telah bertolak dari 'Arafat, berdzikirlah kepada Allah di Masy'arilharam. Dan berdzikirlah (dengan menyebut) Allah sebagaimana yang ditunjukkan-Nya kepadamu dan sesungguhnya kamu sebelum itu benar-benar termasuk orang-orang yang sesat.” (Q.S Al-Baqarah: 198).

Diterangkan pada kitab tafsir Al-Jami' liahkam al-Qur'an wa al-Mubayyin Lima Tadammanahu Min as-Sunnah wa Ayi al-Furqan yang ditafsirkan yakni Sebelumnya ada di antara kalian yang merasa bersalah jika melakukan perniagaan dan mencari rezeki pada musim haji. Sebenarnya, kalian tidak berdosa melakukan hal itu. Maka berniagalah dengan cara-cara yang disyariatkan, carilah karunia dan nikmat Allah. Apabila para haji telah beranjak dari Arafah setelah melakukan wukuf dan mereka tiba di Muzdalifah pada malam Idul Adha, maka hendaknya mereka berzikir kepada Allah di al-Masy'ar al-Harâm, di bukit Muzdalifah. Hendaknya mereka memperbanyak tahlîl (membaca "lâ ilâha illâ Allâh"), talbiyah (membaca "labbayka Allâhumma labbayk", dst.) dan takbîr (membaca "Allâhu Akbar"). Agungkan dan pujilah nama Tuhanmu yang telah memberi hidayah untuk memeluk agama yang benar dan melakukan ibadah haji. Sebelumnya mereka itu berada dalam kesesatan (Al-Qurtubi ,620 H).

Maka dari itu dirancanglah aplikasi *Mobile* yang dapat membantu perniagaan pada model transportasi jarak dekat yang sesuai dengan keadaan jarak tempuh dan medan yang ada. Dimana yang menjadi parameter utama

dalam penentuan harga dari penggunaan jasa transportasi dapat diambil dari jarak dan medan yang dilalui oleh penyedia jasa transportasi sewaktu mengantar pengguna jasa transportasi agar tidak terjadi pengurangan takaran atau hak dari pengguna jasa. Diterapkannya hal ini agar terjadi keadilan pada takaran dan hak, dikarenakan bila menggunakan cara lama maka akan memakan waktu yang cukup lama untuk bernegosiasi harga dan belum tentu akan sepakat pada satu harga diawal karena biasanya ada biaya tambahan bila penyedia jasa transportasi meminta tambahan biaya diakhir.

1.2 Pernyataan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka pertanyaan penelitian yang akan diangkat pada penelitian ini adalah sebagai berikut:

1. Seberapa akurat hasil perkiraan jarak dan medan menggunakan metode haversine sehingga dapat bersesuaian dengan keadaan jarak dan medan yang sebenarnya?
2. Seberapa akurat hasil terapan tarif berdasarkan dari uji perbandingan dengan ketentuan pemenuhan sehingga tidak merugikan pengguna?

1.3 Tujuan Penelitian

Adapun tujuan dilakukannya penelitian ini adalah sebagai berikut:

1. Mengukur jarak dan kontur kemiringan medan dengan sistem yang terotomasi menggunakan metode haversine sehingga sesuai dengan keadaan sebenarnya.
2. Mengukur terapan estimasi tarif sehingga sesuai dengan keadaan jarak dan medan yang telah diperkirakan.

1.4 Batasan Penelitian

Pembahasan pada penelitian ini dibatasi oleh beberapa hal berikut:

1. Parameter yang digunakan jarak dan kontur kemiringan medan.
2. Aplikasi yang dibangun diperuntukan untuk digunakan oleh penumpang dan ojek.
3. Terapan metode dilakukan dalam estimasi biaya jarak tempuh yang dihitung berdasarkan jarak dan medan yang akan ditempuh.
4. Implementasi kontur kemiringan daerah yang diambil menggunakan letak geografis dari Indonesia timur.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini antara lain:

1. Menghasilkan aplikasi ojek untuk Indonesia wilayah timur yang sesuai dengan kebutuhan yang ada.
2. Pengoptimalan perkiraan jarak dan medan dalam sistem yang terotomasi sehingga sesuai dengan keadaan sebenarnya, serta tarif yang adil.
3. Adanya hasil yang didapatkan dari penggunaan metode *Haversine* dan *Location Based Service* dapat menjadi inovasi baru didunia digitalisasi.

BAB II

STUDI LITERATUR

2.1 *Location Based Service*

Location Based Service (LBS) adalah layanan yang dapat diakses dengan perangkat *Mobile* melalui jaringan seluler dan memanfaatkan kemampuan untuk menggunakan lokasi pengguna (Virrantaus dkk, 2001). Bila dilihat dari sudut pandangan pengguna didefinisikan *LBS* sebagai layanan bagi pengguna ponsel yang mengambil posisi pengguna saat ini kedalam akun pengguna (Zipf, 2002). dan didefinisikan *LBS* sebagai layanan yang mengintegrasikan perangkat *Mobile* lokasi atau posisi dengan informasi lain untuk memberikan nilai tambah pengguna (Schiller dkk, 2004). Di lihat dari sudut pandang lebih yang berorientasi pada sistem, layanan tersebut juga didefinisikan sebagai hubungan antara *web service* dan perangkat *Mobile* yang setiap perangkat *Mobile* akan diberi akun untuk mengetahui posisi penggunaanya (Ait-Cheik-Bihi dkk, 2011).

2.1.1 *Global Positioning System*

Global Positioning System (GPS) adalah suatu sistem navigasi satelit yang berfungsi untuk menentukan lokasi, kecepatan, dan arah, melalui *signal* yang diterima lebih dari 24-32 satelit yang berorbit 20000 km (11000 mil laut) di atas bumi. Prinsip dasar di balik GPS adalah pengukuran jarak (*distance/range*) antara satelit dan *receiver* dari transmisi *signal radio*. Sebuah GPS *receiver* memerlukan empat atau lebih satelit untuk menghasilkan jarak mereka, dan menggunakan informasi ini untuk menyimpulkan lokasi mereka. Operasi ini disebut

triangulation. Satelit-satelit GPS mentransmisikan *signal radio* ke bumi yang mengandung informasi tentang satelit, contohnya lokasi dan waktu saat itu, menginformasikan pengguna secara tepat di mana satelit berada dalam orbit mereka sementara GPS *receivers* secara pasif menerima *signal* satelit tersebut. Semua satelit-satelit GPS menyinkronisasi operasi-operasi agar *signal-signal* yang berulang ini ditransmisikan pada waktu yang bersamaan.

Pengimplementasian *point end-to-end location* atau lokasi yang memanfaatkan kemampuan *based location* pada *smartphone* yang memanfaatkan fitur dari GPS. Metode ini pun digunakan untuk menentukan titik awal dan titik tujuan melalui pengambilan lat-lang dari pengguna *smartphone*. Pengindikasian lokasi pada pengguna akan dapat dilakukan apabila perangkat yang digunakan mendukung penggunaan fitur GPS untuk menarik titik awal lokasi sehingga nantinya akan dapat ditarik garis sejajar dengan rute yang akan dilalui menuju titik yang dituju. Dimana titik itu akan di *tag* atau akan dijadikan patokan utama sehingga jarak dan medan yang akan dilalui dalam perjalanan menuju titik tujuan dapat diolah dengan mengambil *value* dari rute sebagai parameter pengindikasian berapa jauh jarak yang akan dilalui dan seperti apa medan yang akan dilalui.

Ilustrasi lebih jelasnya ada pada gambar 2.1 dan gambar 2.2



Gambar 2.1 Area pertama, kedua dan ketiga (Odijk D, 2002)



Gambar 2.2 *Triangulation* (Odijk D, 2002)

Rumus yang digunakan untuk menghitung jarak adalah rumus *Great Circle Distance*, yaitu

$$\begin{aligned} \text{Great Circle Distance} = & \text{RadiusBumi} * \text{ACOS} (\text{COS}(\text{RADIAN}(90-(\text{Lat1}*24))) * \\ & \text{COS}(\text{RADIAN}(90-(\text{Lat2}*24))) + \text{SIN}(\text{RADIAN}(90-(\text{Lat1}*24))) * \\ & \text{SIN}(\text{RADIAN}(90-(\text{Lat2}*24))) * \text{COS}(\text{RADIAN}(24*(\text{Long1}-\text{Long2}))))..... (1) \end{aligned}$$

Dimana :

Lat1 adalah *latitude* nilai 1, dalam DD:MM:SS;

Long1 adalah *longitude* nilai 1, dalam DD:MM:SS;

Lat2 adalah *latitude* nilai 2, dalam DD:MM:SS;

Long2 adalah *longitude* nilai 2, dalam DD:MM:SS;

Radius Bumi adalah radius bumi (3963 mil atau 6377 kilometer).

2.2 Haversine

Rumus *Haversine* adalah persamaan penting dalam proses navigasi, *Haversine Formula* memberikan jarak lingkaran besar (radius) antara dua titik pada permukaan bumi berdasarkan bujur dan lintang. Penggunaan formal ini mengasumsikan pengabaian efek elipsoidal (diasumsikan bumi tidak bulat sempurna melainkan lebih mendekati bentuk telur dengan permukaan yang tidak rata). Untuk dua titik pada bola (dari radius R) dengan garis lintang dan ϕ_1 and ϕ_2 , pemisah lintang $\Delta\phi = \phi_1 - \phi_2$, dan bujur pemisah $\Delta\lambda$, dimana sudut dalam radian, d jarak antara dua titik (sepanjang lingkaran besar dari bola melihat jarak bola) berhubungan dengan lokasi oleh rumus sebagai berikut :

$$\text{Haversin} \left(\frac{d}{R} \right) = \text{haversin}(\Delta\phi) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\Delta\lambda) \dots \dots \dots (2)$$

Dimana :

$$\text{Haversine}(\phi) = \sin^2(\phi/2) = (1 - \cos(\phi))/2$$

ϕ_1 = garis lintang (*latitude*) dari titik 1.

d = jarak dua titik (antara posisi pengguna dan koordinat yang dituju).

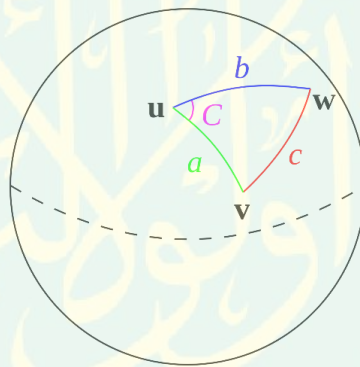
ϕ_2 = garis lintang (*latitude*) dari titik 2.

R = jari – jari bola (radius bumi).

$\Delta\lambda$ = pemisah bujur (*longitude*).

Sudut pada rumus menggunakan satuan radian untuk menggunakan fungsi trigonometri. Rumus *Haversine* digunakan pada penelitian ini untuk melakukan perhitungan jangkauan radius pencarian rute yang dituju oleh pengguna. Rumus ini juga digunakan untuk substitusi perhitungan jarak posisi lokasi tujuan dan posisi pengguna. Pengimplementasian *Haversine* akan digunakan untuk menghitung jarak antara dua titik koordinat dari GPS. Dalam hal ini adalah titik koordinat posisi pengguna dengan posisi koordinat yang dituju, sehingga penggunaan *Haversine* menjadi kunci utama untuk menentukan jarak antara posisi pengguna dengan posisi lokasi tujuan berdasarkan radius.

Ilustrasi penggunaan *Haversine* dapat dilihat pada gambar 2.3



Gambar 2.3 Segitiga bola diselesaikan dengan hukum *Haversine Formula*
(Gintoro dkk, 2010)

Jika panjang dari ketiga sisi adalah a (u ke v), b (u untuk w), dan c (v ke w), dan sudut c sebaliknya adalah C , maka hukum *Haversine* yang akan didapatkan adalah $\text{Haversin}(c) = \text{haversin}(a-b) + \sin(a)\sin(b)\text{haversine}(c)$.

Maka dari rumus tersebut didapatkan rumus yang akan diterapkan pada penelitian ini seperti berikut.

$R = \text{Radian bumi} = 6371 \text{ km}$.

$$\Delta lat = lat2 - lat1$$

$$\Delta lang = lang2 - lang1$$

$$a = \sin^2(\Delta lat/2) + \cos(lat1) * \cos(lat2) * \sin^2(\Delta lang/2)$$

$$c = 2 * a \tan^2(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c \dots\dots\dots (3)$$

Dari perhitungan di atas, untuk menentukan jarak paling dekat harus terdapat pembanding. Jadi harus terdapat titik lokasi maka akan ditemukan jarak. Untuk membatasi ruang peta, digunakan radius agar wilayah yang ditampilkan tidak melebar. Untuk jalur jalan yang sebenarnya sesuai kondisi memanfaatkan *Google Maps (API)*, lalu akan ditampilkan *marker* dimana rute akan diberi warna biru. Untuk perhitungan *query* akan dilakukan pada *cloud server* dan hasilnya akan berupa objek data *JSON* yang akan diakses dengan menggunakan *source code* cara *etrofit* atau *async*.

2.3 One-Way Anova

ANOVA (Analysis of Variance) atau sering juga diistilahkan sebagai uji sidik ragam, dikembangkan oleh Ronald Fisher. Prinsip pengujiannya adalah menganalisis variabilitas atau keragaman data menjadi dua sumber variasi, yaitu variasi dalam kelompok (*within*) dan variasi antar kelompok (*between*). Bila variasi *within* dan *between* sama maka rata-rata yang dihasilkan tidak ada perbedaan, sebaliknya bila hasil perbandingan kedua varian tersebut menghasilkan nilai lebih dari 1, maka rata-rata yang dibandingkan menunjukkan adanya perbedaan.

Beberapa asumsi dasar yang mesti dipenuhi pada uji ANOVA adalah:

- a. Data sampel yang digunakan berdistribusi normal atau dianggap normal.

- b. Populasi tersebut memiliki varian yang homogen.
- c. Sampel tidak berhubungan satu dengan lain (independen), sehingga uji ANOVA tidak bisa digunakan untuk sampel berpasangan (paired).

Menghitung jumlah kuadrat total dengan menggunakan rumus.

$$SS_T = \sum (X_{ij})^2 - \frac{(\sum T_j)^2}{n} \dots\dots\dots (4)$$

Setelah mendapatkan kuadrat total maka tahap selanjutnya menghitung jumlah kuadrat perlakuan dengan rumus.

$$SS_p = \sum \frac{(\sum T_j)^2}{n_j} - \frac{(\sum T_j)^2}{n} \dots\dots\dots (5)$$

2.4 Penelitian Terkait

Penggunaan pencarian rute pada sistem transportasi tidak bisa disamakan dengan pencarian rute pada lingkup yang luas. Proses pencarian rute pada sistem transportasi publik perlu memperhatikan bahwa beberapa kendaraan hanya dapat melewati rute-rute tertentu yang telah ditentukan sebelumnya, dan penumpang tidak dapat meminta pengemudi untuk mengganti rutanya (Chao dkk, 2001). Algoritma yang dapat digunakan untuk menyelesaikan masalah perencanaan rute terbaik pada transportasi pernah diteliti oleh Hsiao dkk, Geissberger, Liu dkk, dan Chao-Lin Liu.

Hsiao dkk menawarkan algoritma solusi berdasarkan teknik optimasi koloni semut (*ant colony optimization*) untuk mencari rute terpendek dari suatu lokasi asal ke lokasi tujuan pada peta dengan mempertimbangkan kondisi lalu lintas (Hsiao, 2004). *Ant Colony Optimization* (ACO) adalah representasi grafik berdasarkan meta-heuristik algoritma *evolusioner*. ACO telah berhasil digunakan untuk memecahkan banyak masalah optimasi kombinasi yang

berbeda. Ide utama ACO adalah untuk model masalah sebagai jalur biaya minimum mencari masalah dalam grafik. ACO terdiri dari banyak semut buatan, dan semut buatan ini berjalan melalui grafik untuk mencari jalan terpendek dari grafik. Semut buatan memiliki perilaku yang sangat sederhana, dan biasanya, itu hanya mampu menemukan jalan berkualitas rendah sangat dengan sendirinya. Jalur yang lebih baik ditemukan oleh kerjasama global antara semua buatan semut di koloni.

Tujuan dari perencanaan rute yang terbaik adalah untuk menemukan rute terpendek dari *node* asli yang diinginkan ke *node* tujuan yang diinginkan. Objek dari penelitian tersebut adalah untuk mendapatkan jalan terpendek untuk Masalah perencanaan jalur yang mempertimbangkan paket mencegah melewati berat memuat jalan dan menghindari *delay transportasi*. Pertama, adalah perlu untuk menentukan lingkup peta. *Node* dan rute dari peta untuk pengujian dapat diatur secara manual, atau dihasilkan oleh komputer secara acak. Dalam rangka untuk mengevaluasi fleksibilitas dan kinerja dari algoritma yang diusulkan, hsiao menggunakan *random* menghasilkan peta untuk mengujinya. Gambar 2.4 menunjukkan peta pembangkit acak. Setiap *node* memiliki koordinat mutlak sendiri, dan panjang setiap rute dapat dihitung dengan koordinat dari aslinya simpul terminal dan *node* tujuan terminal. Pemuatan *parameters* LD dari masing-masing rute harus diatur secara manual atau ditetapkan sebagai 0 secara *default*. Pada langkah pertama dari simulasi, itu harus memilih *node* asli yang diinginkan dan *node* tujuan.



Gambar 2.4 *Random Generated Map* (Hsiao, 2004)

Banyak masalah perencanaan rute yang dapat diselesaikan dengan menghitung jarak terpendek menggunakan pemodelan yang cocok, graf berbobot yang merepresentasikan jaringan rute. Pada rute transportasi, perlu diperhitungkan waktu yang diperlukan untuk melakukan *transfer* antar rute (Geisberger, 2011). Liu dkk menggunakan matrik konektivitas (*connectivity matrices*) yang merupakan pengembangan dari matrik kedekatan (*adjacency matrices*) untuk menyelesaikan masalah batasan rute pada transportasi. Strategi lain yang ditawarkan adalah menggunakan hubs (Chao dkk, 2001). Dengan mempertimbangkan bahwa *transfer* antar rute pada transportasi memerlukan lebih banyak waktu dan biaya, digunakan matrik transisi (*transition matrices*) untuk memperhitungkan jumlah *transfer* yang dibutuhkan oleh penumpang (Chao, 2002).

Beberapa metode algoritma yang dapat digunakan untuk mencari rute terbaik pada transportasi banyak dikemukakan oleh para peneliti. Pencarian rute terbaik dan rute alternatif menggunakan algoritma Dijkstra pernah dilakukan oleh Haryanto dengan mengambil studi kasus Kota Surakarta. Aplikasi yang dibuat berbasis WAP menggunakan PHP dan MySQL. Penggunaan teknologi

WAP memungkinkan aplikasi ini dapat diakses melalui telepon seluler yang mempunyai fasilitas WAP. Penelitian ini cocok untuk orang yang bepergian menggunakan kendaraan pribadi (Haryanto, 2008).



BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai beberapa hal, yaitu tahapan penelitian yang akan dilakukan, kebutuhan sistem yang akan dibuat dan penyelesaian masalah dalam penentuan rute transportasi roda dua untuk wilayah Indonesia bagian timur dengan *sample* data daerah NTB dengan menggunakan kombinasi antara *Location Based Service* dan *Haversine* yang diimplementasikan dengan menggunakan *Mobile Cloud Programming*. Tahapan metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur metode penelitian

3.1 Kondisi Lingkungan Penelitian

Penelitian ini dilakukan di Nusa Tenggara Barat dengan mengambil objek penelitian yaitu Angkutan roda dua yang biasa dikenal dengan sebutan ojek. Angkutan roda dua di Nusa Tenggara Barat yang menghubungkan daerah

– daerah plosok atau akses jalan yang masih belum dilayak. Ada 3 parameter utama yang akan menjadi rujukan utama dari daerah NTB yang memiliki letak geografis dan luas wilayah berbeda-beda. Berikut pada tabel 3.1 di paparkan mengenai letak/luas daerah, jarak wilayah tiap kabupaten, ketinggian wilayah secara geografis.

Tabel 3.1 Letak daerah, jarak wilayah tiap kabupaten, luas wilayah dan ketinggian wilayah (Badan Pusat Statistik Nusa Tenggara Barat, 2016)

Wilayah Kabkota	Tinggi Wilayah dan Ibukota	Luas Daerah Nusa Tenggara Barat		Jarak Ibukota Kabkota ke Ibukota Provinsi	
	Tinggi (mdpal)	Luas (km ²)	Persentase (%)	Jarak ke Ibukota Provinsi (km)	
	2016	2016	2016	2015	2016
Kabupaten Lombok Barat	15	1053.9	5.23	20.3	20.3
Kabupaten Lombok Tengah	107	1208.4	6	27	27
Kabupaten Lombok Timur	166	1605.6	7.97	52	52
Kabupaten Sumbawa	18	6644	32.97	179	179
Kabupaten Dompu	30	2324.6	11.53	377	377
Kabupaten Bima	17	4389.4	21.78	414	414
Kabupaten Sumbawa Barat	11	1849	9.17	111.7	111.7
Kabupaten Lombok Utara	12	809.53	4.02	50.5	50.5
Kota Mataram	27	61.3	0.3	0	0
Kota Bima	21	207.5	1.03	439	439

3.2 Perancangan System

Perancangan *System* merupakan dasar dari pembuatan suatu sistem yang sesuai kebutuhan dan terotomasi. Untuk melakukannya memerlukan tahap-tahapan dalam perencanaan *system* yang dibagi menjadi 3 yaitu:

1. Akuisisi Data, bertujuan untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya.
2. Perencanaan Proses
3. Perancangan *System* atau Desain *System*

3.2.1 Akuisisi Data

Data yang digunakan adalah data primer dan data sekunder, dengan rincian seperti yang dijelaskan pada bagan penjelasan 1 dan 2.

1. *Data Training*

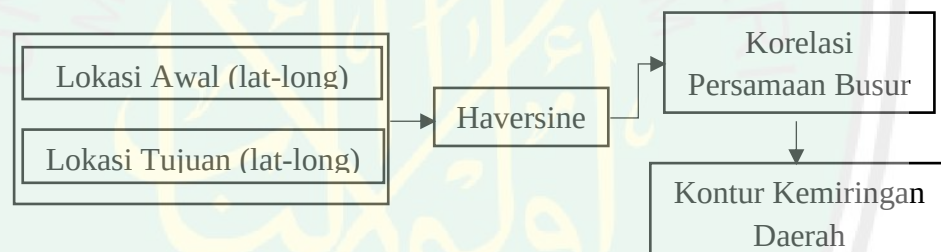
Data Training yang digunakan adalah data hasil observasi lapangan berupa data dalam bentuk koordinat letak lokasi ataupun data *lat-lang*. Data ini akan dicocokkan dengan data yang sudah ada di *Website* resmi Badan Pusat Statistik Nusa Tenggara Barat yang telah menyediakannya secara terbuka. Hasilnya akan dijadikan acuan dalam membentuk perhitungan matrik *Trigonometric* dalam pengimplementasian *Haversine*.

2. *Data Testing*

Sedangkan data *Testing* yang digunakan berupa jenis pengkondisian medan dari rute yang dilewati. Data ini berisikan pengkondisian medan dengan berbagai trayek yang berbeda-

beda. Data ini nantinya akan dijadikan sebagai acuan dalam perhitungan dan selanjutnya menjadi bahan dalam implementasi *Haversine*.

Dari beberapa data yang cukup banyak tersebut, untuk menguji metode dalam penelitian ini dibutuhkan 3 jenis pengkondisian medan sehingga memudahkan dalam proses pengujian aplikasi yang dibangun dalam penelitian ini. Data yang dibutuhkan mengenai rute angkutan roda dua, daerah yang dilewati yang memiliki koordinat dalam menentukan kondisi medan daerah-daerah yang dilewati. Adapun penyimpanan data dan tahapan memprosesnya digambarkan ke dalam gambar 3.2



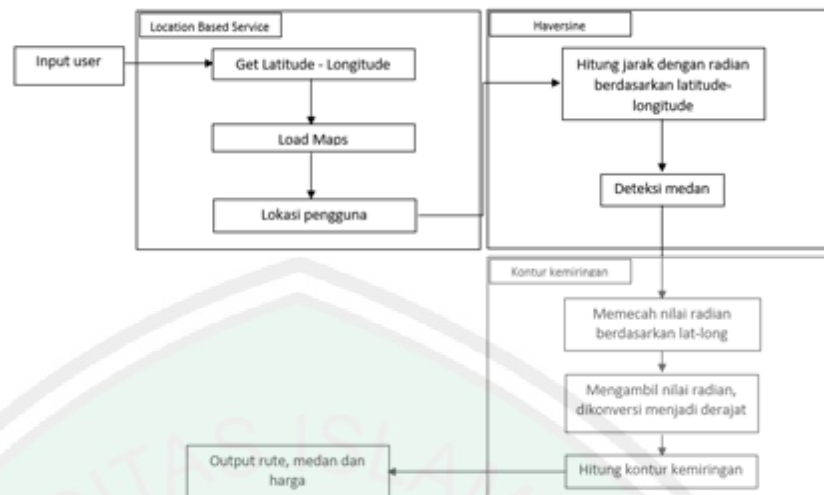
Gambar 3.2 Blok diagram akuisisi data

3.2.2 Perancangan Proses

Perancangan Proses dilakukan untuk mempermudah implementasi, pengujian, serta analisis. Sistem akan dibangun menggunakan *platform android* beserta antamuka yang dapat memudahkan pengguna untuk menggunakan sistem yang akan dibangun.

3.2.2.1 Desain System

Desain sistem untuk pengimplementasian kombinasi antara *LBS* dan *Haversine* dapat dilihat pada gambar 3.3.



Gambar 3.3 Desain sistem

Dalam sistem yang akan dibangun, data yang digunakan adalah data koordinat atau *lat-long* dari lokasi awal dan tujuan maupun pengambilan data yang sudah disediakan. Serta inputan berupa daerah atau jalan tujuan sehingga pada nantinya akan melalui proses pencarian rute sehingga akan menghasilkan rekomendasi angkutan roda dua yang terdekat dengan lokasi pengguna. Berikut merupakan tahapan-tahapannya:

A. *Input User*

Pada langkah awal, adalah proses penginputan data tujuan dari pengguna. Data yang diinputkan berupa tempat yang ingin dituju, atau nama jalan dan nama tempat. Kemudian data inputan dari pengguna akan konversi kedalam bentuk koordinat dan data koordinat tersebut akan diparsing kedalam proses selanjutnya dengan bentuk *variable* baru. Pada dasarnya pengguna diberi keleluasaan dalam penginputan data dikarenakan data tersebut

akan diolah dengan menggunakan metode *Auto Complete Location* yang merupakan fitur pendukung dari *Google*.

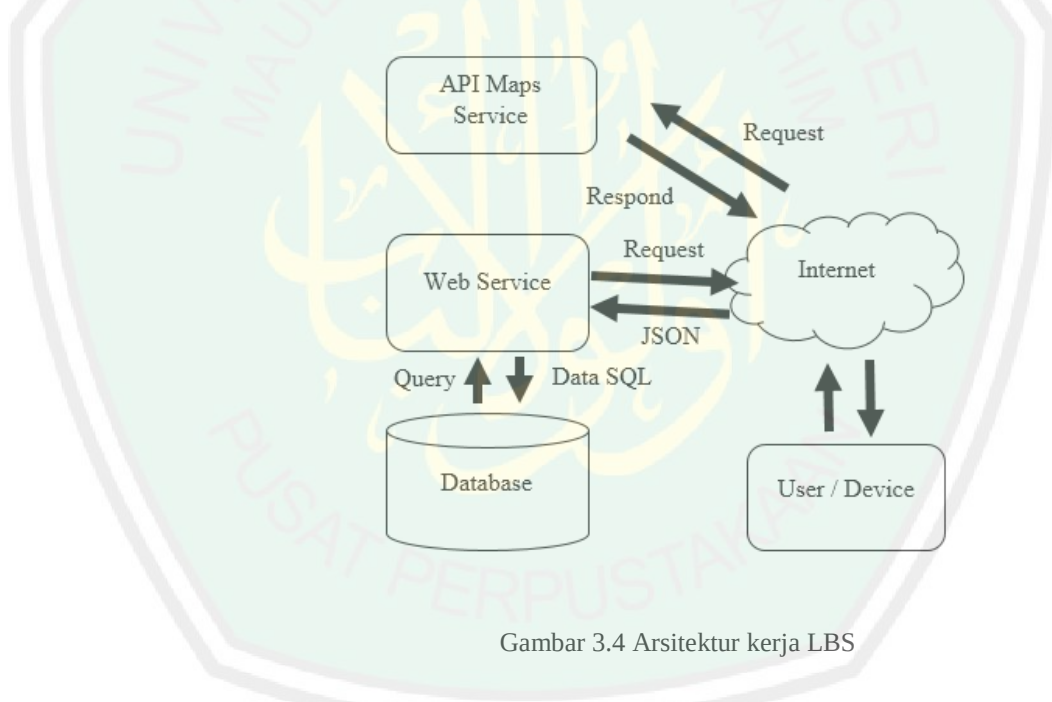
B. LBS (*Location Based Service*)

LBS app biasanya ketika pertama diaktifkan fitur *location* langsung akan mencari tahu lokasi pengguna saat ini. Setelah Lokasi diketahui maka melalui *internet* aplikasi akan melakukan *request* ke *Google Maps* sebagai penyedia *API*. *Request* yang dilakukan berupa *request* peta berdasarkan lokasi *user* saat ini, dan ketika *request* itu berhasil diterima, maka *Google Maps* akan segera merespon aplikasi dengan memberikan akses peta kepada aplikasi.

Selain itu Ketika lokasi berhasil diketahui, maka melalui internet juga aplikasi akan melakukan *request* ke *Web Service* yang dibuat sendiri. *Web Service* berguna untuk mengambil data pada *Database*, *request* dari aplikasi tadi diubah menjadi *Query* oleh *Web Service*. Ketika database yang berbentuk *SQL* ini berhasil menerima lalu memproses *Query*, maka *Database* akan memberikan data *SQL* sesuai dengan yang diminta oleh *Web Service*. Sekali lagi *Web Service* mengubah data *SQL* yang diterimanya menjadi *Json* format yang akan lebih mudah dimengerti oleh aplikasi *Smartphone*. Dan ketika *Json* yang dikirimkan *Web Service* diterima oleh aplikasi, maka aplikasi akan menampilkan data sesuai dengan lokasi pengguna berada. *Map* dan

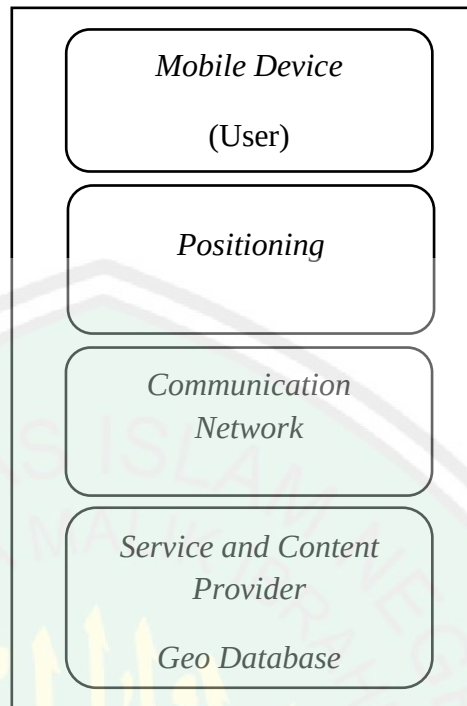
Layanan berbasis lokasi menggunakan garis lintang dan garis bujur untuk menentukan letak geografis, namun sebagai pengguna membutuhkan alamat atau posisi *Real Time* bukan nilai garis lintang dan garis bujur.

Android menyediakan *Geocoder* yang mendukung *Forward* dan *Reverse Geocoding*. Menggunakan *Geocoder*, posisi yang berupa garis lintang dan garis bujur akan dikonversi menjadi alamat dunia nyata ataupun sebaliknya. Adapun arsitektur cara kerja *LBS* digambarkan pada gambar 3.4



Gambar 3.4 Arsitektur kerja LBS

Adapun komponen pendukung utama dalam teknologi layanan Berbasis Lokasi tertera pada gambar 3.5



Gambar 3.5 Komponen LBS

C. Load Maps

Langkah selanjutnya adalah *Load Maps*. *Load Maps* yang dipakai dengan menggunakan *Google Maps*. *Google Maps* adalah peta *virtual* yang disediakan gratis oleh *Google* dan bisa diakses online oleh siapapun melalui situs *Google Maps* menyediakan banyak fitur, salah satunya adalah pencarian rute dari suatu tempat ke tempat yang lain (N. Azmi, 2013). *Google Maps* juga bisa diakses melalui *Mobile Phone*. Apalagi dengan didukung oleh GPS dari *Mobile Phone*, maka aplikasi dari *Google Maps* ini pun akan sangat terasa manfaatnya antara lain sebagai *location tracking* (Rizky, 2012).

Selain itu, *Google Maps* juga menyediakan *API* (*Application Programming Interface*) tidak berbayar untuk diintegrasikan dengan aplikasi lain. Untuk gambar yang ditampilkan dari *Google Maps* itu sendiri bukanlah gambar yang diperbarui secara *real-time*, melainkan gambar yang telah berbulan-bulan usianya. Akan tetapi terkadang gambar yang ditampilkan adalah gambar terbaru yang biasanya dikarenakan adanya kejadian-kejadian yang sangat khusus. Hal ini sangat mungkin dilakukan karena meskipun *Google* menggunakan kata satelit, beberapa gambar resolusi tinggi yang ditampilkan adalah gambar-gambar *aerial photography* yang diambil dengan menggunakan pesawat yang mengudara pada ketinggian 800-1500 kaki. Selain itu, beberapa gambar tidak sama tingkat resolusinya. Biasanya, semakin sedikit populasi suatu daerah, maka semakin kecil pula resolusi gambar di daerah tersebut. Dan terkadang di beberapa daerah gambarnya tertutup oleh awan.

Google Maps API digunakan dalam aplikasi ini agar pengguna dapat mengetahui lokasinya pada peta yang ditampilkan. Selain itu, juga digunakan untuk menghitung jarak antara pengguna dengan lokasi tujuan.

D. Haversine

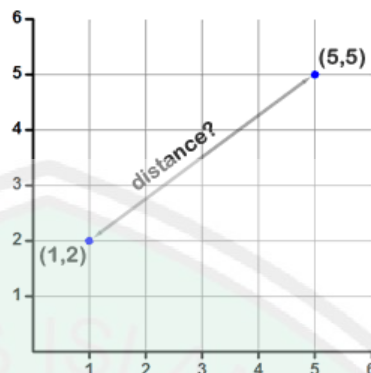
Langkah dalam pengimplementasian *Haversine* dapat dilakukan perhitungan dengan menggunakan fungsi perhitungan

Euclidean distance yaitu perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini berkaitan dengan teorema *pythagoras* dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

1. Langkah perhitungan pada 1 dimensi

Semisal ingin menghitung jarak *Euclidean* 1 dimensi. Titik pertama adalah 7.9 yang bisa diambil dari titik *lat-long* awal, titik kedua adalah -112.6 yang bisa diambil dari titik *lat-long* tujuan. Caranya adalah lakukan operasi pengurangan -112.6 dengan 7.9. Sehingga menghasilkan -120.5. Cari nilai absolut dari nilai -120.5 dengan cara memangkatkannya dengan 2 sehingga mendapat nilai 145.26. Kemudian diakarkan sehingga mendapatkan nilai 120.5. Sehingga jarak *Euclidean* dari 2 titik tersebut adalah 120.5.

2. Langkah perhitungan pada 2 dimensi



Gambar 3.6 Euclidean 2 dimensi

Caranya hampir sama, misalkan titik pertama mempunyai kordinat (1,2). Titik kedua ada di kordinat (5,5). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, (5-1,5-2) sehingga menjadi (4,3). Kemudian pangkatkan masing-masing sehingga memperoleh (16,9). Kemudian tambahkan semuanya sehingga memperoleh nilai $16+9 = 25$. Hasil ini kemudian diakarkan menjadi 5. Sehingga jarak euclideannya adalah 5.

Rumus *Euclid*

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \dots\dots\dots (6)$$

Sehingga dari *Formula* tersebut dapat implementasi menjadi,

$$\text{Jarak} = \sqrt{(\text{lat}_1 - \text{lat}_2)^2 + (\text{long}_1 - \text{long}_2)^2} \dots\dots\dots (7)$$

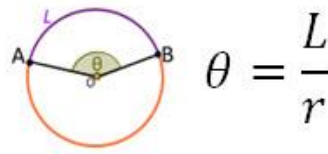
Hasil perhitungan (*Jarak*) masih dalam satuan *decimal degree* (sesuai dengan format longlat yang dipakai) sehingga untuk

menyesuaikannya perlu dikalikan dengan **111.319 km** (1 derajat bumi = 111.319 km).

3. Langkah perhitungan *Haversine*

Kemudian dari penyelesaian yang dilakukan pada perhitungan tahap 2 dengan menggunakan *Euclid* maka dapat diteruskan kedalam perhitungan *Haversine* ditahap 3. Dimana hasil perhitungan *Euclid* ditahap 2 akan menjadi rujukan dalam menjalankan fungsi hitungan *Haversine* ditahap 3. Didalam pengimplementasian *Haversine* akan dibandingkan antara koordinat awal dengan koordinat tujuan yang dari kedua koordinat tersebut akan dihitung jarak dan peletakan rute. Sehingga medan tempuh akan dapat digambarkan melalui pengambilan koordinat yang ada dalam rute yang telah digambarkan.

Ilustrasi pengimplementasian *Haversine* dapat dilihat pada gambar 2.3 pada pembahasan sebelumnya. Pada ilustrasi tersebut didapatkan beberapa parameter a , b , c ialah sebagai jarak yang bersatuan radian/sudut karena berada dalam bidang bola, yang bisa dikorelasikan dengan persamaan busur pada gambar 3.7.



Gambar 3.7 Rumus busur

Kemudian diimplementasikan kedalam persamaan *harvesine*,

$$\text{Haversin}(\emptyset) = \sin^2\left(\frac{\emptyset}{2}\right) = \frac{1 - \cos(\emptyset)}{2} \dots\dots\dots (8)$$

Sehingga dari *Formula* dapat implementasi menjadi,

$$\text{Jarak} = 2r \cdot \arcsin \left\{ \sqrt{\sin^2\left(\frac{\text{lat}_1 - \text{lat}_2}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\text{long}_1 - \text{long}_2}{2}\right)} \right\} \dots\dots\dots (9)$$

Adapun tahapan selanjtunya yakni perhitungan *Euclid* dan *Haversine* dapat dilakukan dengan langkah awal mengambil titik koordinat dari titik awal dan titik tujuan dapat dilakukan dengan cara perhitungan dibawah.

Titik 1 : Kota Bima

Lat : -8.4818619

Long : 118.7739739

Titik 2 : Kota Sape

Lat : -8.5746799

Long : 118.9759335

Jarak aktual yang diperoleh dari *Google Map* adalah 25.00 Km.

Kemudian dibuktikan dengan perhitungan pada *Excel* pada tabel

3.2

Tabel 3.2 Perhitungan *Euclid* dan *Haversine*

perhitungan euclid dan haversine		
Tempat	latitude	longitude
kota bima	-8.4818619	118.7739739
kota sape	-8.5746799	118.9759335
Metode	hasil	
Haversine	24.49000197	km
kontur kemiringan	1,2389	persen derajat

Dengan rumus :

Euclid :

$$\text{Jarak} = \sqrt{((-8.4818619) - (-8.5746799))^2 + (118.7739739 - 118.9759335)^2} \cdot 111.319 = 24.74260074$$

Haversine :

$$\begin{aligned} \text{Jarak} &= 6371,1 \cdot \left(\arcsin \left(\sqrt{\sin^2 \left(\frac{(-8.4818619) - (-8.5746799)}{2} \right) + \cos(-8.4818619)} \right. \right. \\ &\quad \left. \left. \cdot \cos(-8.5746799) \cdot \sin^2 \left(\frac{118.7739739 - 118.9759335}{2} \right) \right)} \right) \\ &= 24.49000197 \end{aligned}$$

Mengambil nilai radian pada haversine untuk dikonversi ke

derajat :

$$K_{\text{tertinggi}} = ((180^0(-8.4818619) - 180^0(-8.5746799))/2)^2 = 6.5$$

$$K_{\text{terendah}} = ((180^0(118.7739739) - 180^0(-118.9759335))/2)^2 = 3.1$$

Kontur kemiringan :

kontur tertinggi – kontur terendah / skala sebenarnya * 100 %

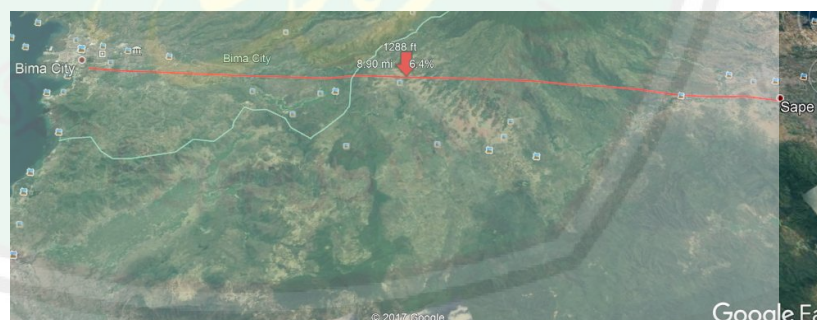
(satuan persen dapat diganti dengan satuan derajat)(10)

maka dapat dihitung :

Kontur kemiringan = $1.233 - 1.239 / 100.000 * 1\% = 1,2389\%$

Kontur kemiringan = $1.233 - 1.239 / 100.000 * 1^0 = 1,2389^0$

Dari perhitungan yang dilakukan dapat dihasilkan dalam bentuk persen atau derajat. Hasil yang telah didapatkan tersebut dapat diambil dalam bentuk persen atau derajat dengan cara mengubah satuannya. Dan dapat dilihat pada google earth tingkat kontur kemiringan pada daerah yang akan dilewati seperti pada gambar 3.8.



Gambar 3.8 Kontur kemiringan pada *google earth*

Dari jarak yang telah didapatkan akan digambarkan kedalam rute yang sesuai dengan kebutuhan. Kemudian dengan penggambaran rute akan didapatkan medan yang akan dilalui

oleh pengguna nantinya, sehingga dapat memberikan rincian akumulasi lokasi, medan dan biaya dari rute yang telah muncul. Dimana pengguna akan dapat mengetahui harga yang ada telah sesuai dengan kondisi jarak dan medan lokasi yang dituju.

E. Proses Load Rute Dan Medan Dengan Autocomplete Maps

Layanan *place autocomplete* dalam *Google Places API for Android* memberikan prediksi tempat sebagai *respons* terhadap inputan penelusuran oleh pengguna. Saat pengguna mengetikkan nama tempat atau lokasi maka *place autocomplete* akan memberikan alternatif untuk tempat seperti bisnis, alamat, dan tempat-tempat lain. Kemudian *API* mengembalikan *Autocomplete PredictionBuffer* dalam *Pending Result*. *Autocomplete Prediction Buffer* yang didalamnya berisi daftar objek *Autocomplete Prediction* yang menyatakan tempat-tempat yang diprediksikan. *Buffer* mungkin kosong, jika tidak ada tempat yang diketahui sesuai dengan inputan pengguna dan kriteria filter. Untuk setiap tempat yang diprediksi bisa menggunakan metode berikut untuk mengambil detail tempat metode yang dapat diimplementasikan yakni *getFullText (CharacterStyle matchStyle)*, *getPrimaryText (CharacterStyle matchStyle)*, *getSecondaryText (CharacterStyle matchStyle)*, *getPlaceId ()*, *getPlaceTypes ()*.

Dari pengimplementasian *autocomplate maps* pada proses penggambaran rute - rute jalan yang dilalui. Kemudian dapat dilihat kevalidan rute atau jaringan jalan tersebut dengan mencocokkan data jaringan jalan yang ada diIndonesia yang dikeluarkan oleh beberapa lembaga yang berwenang dibidang pemetaan. Adapun lembaga yang dapat diambil datanya dala bidang pemetaan diantaranya seperti Badan Pusat Statistik (BPS) ditiap lingkup daerah diIndonesia yang telah mendukung diadakannya pendataan dalam bentuk data sheet, *Open Street Maps* (OSM) yang memiliki kemampuan dalam bidang *survey* serta pemetaan bumi memanfaatkan *tools* dari Arcgis dan Rupa Bumi Indonesia (RBI) yang bergerak dalam pemetaan rupa bumi yang ada diIndonesia baik dari letak wilayah sampai dengan jaringan jalan dengan memanfaatkan *tools* dari ESRI yang menghasilkan data dalam bentuk *shape maps*.

Data *Open StreetMap* diterbitkan dibawah lisensi oleh *Creative Commons* dengan tujuan mempromosikan penggunaan bebas dan penyebaran luas data *Street Map*. Pada September 2012, lisensinya diubah menjadi *Open Database License* (ODbL) dari *Open Data Commons* (ODC) agar izin kepemilikan datanya lebih jelas. Semua data yang dimasukkan ke proyek ini harus memiliki lisensi yang kompatibel dengan *Open Database License*. Data-data tersebut meliputi informasi tanpa hak cipta,

domain umum, atau lisensi lain. Para kontributor menyetujui aturan yang mensyaratkan keserasian (kompatibilitas) dengan lisensi terbaru. Aturan tersebut meliputi pemeriksaan lisensi data pemerintah agar bisa ditentukan keserasian lisensinya. Perangkat lunak yang digunakan dalam proses produksi dan presentasi data *Open Street Map* berasal dari berbagai proyek dan masing-masing memiliki lisensi yang berbeda. *Software* yang dipakai pengguna untuk menyunting peta dan melihat log perubahan didukung oleh Ruby on Rails. *Software* ini juga menggunakan PostgreSQL untuk menyimpan data pengguna dan menyunting metadata. Peta bakunya diproduksi oleh Mapnik, disimpan dalam PostGIS, dan didukung oleh modul Apache bernama *mod_tile*.

Peta RBI adalah Peta Topografi yang dikeluarkan dalam cakupan wilayah meliputi daerah - daerah di Indonesia. Untuk pembuatan peta tematik pada dasarnya digunakan sebagai peta topografi resmi dari suatu Negara. Umumnya peta dasar tersebut dibuat berdasarkan *survey* lapangan atau fotogrametris. Peta RBI biasanya digunakan sebagai dasar pembuatan peta tematik, diperlukan data-data topografi tersebut untuk agar data - data tematis dapat digambarkan. Biasanya jenis peta ini digunakan untuk keperluan peta tematik dalam memperhatikan batas- batas wilayah sangat terperinci.

F. Output

Pada aplikasi yang telah dibangun memiliki *output* biaya yang diberlakukan kepada pengguna. Adapun yang menjadi pedoman dalam menentukan biaya, diambil dari beberapa aturan yang berlaku diantaranya permenhub no 26 dan no 108 tahun 2017 serta perda no 4 tahun 2013. Pada permenhub no 26 tahun 2017 tentang penetapan tarif batas atas dan batas bawah pada angkutan umum. Pada permenhub no 108 tahun 2017 tentang penyelenggaraan angkutan orang dengan kendaraan bermotor umum tidak dalam trayek, diberlakukan tarif batas atas dan batas bawah seperti pada permenhub no 26 serta menyesuaikan dengan perda yang berlaku di tiap daerahnya masing - masing baik itu untuk beberapa model angkutan online yang ada. Untuk cakupan perubahan tarif diterapkan seperti kondisi yang ada pada tabel 3.3.

Tabel 3.3 Perubahan tarif harga

No.	perubahan tarif harga		
	Kondisi	harga normal	tambahan harga
1	Cerah	2.500 - 3.700	-
2	hujan ringan	2.500 - 3.700	Rp.100
3	hujan sedang	2.500 - 3.700	Rp.150
4	hujan berat	2.500 - 3.700	Rp.175
5	pagi/siang/petang	2.500 - 3.700	-
6	malam	2.500 - 3.700	Rp.500-1,000

Dan berdasarkan aturan - aturan yang berlaku maka biaya atau tarif pada aplikasi ini berada pada kisaran yang berbeda sesuai

dengan tarif batas bawah yang diberlakukan oleh pemerintah pusat maupun daerah.

Tabel 3.4 Batas tarif bawah & atas

No.	wilayah cakupan	mobil/ taksi online		sepeda motor online	
		tarif batas bawah	tarif batas atas	tarif batas bawah	tarif batas atas
1	jawa, sumatra & bali	3.500	6.000	1.000	4.000
2	kalimantan, nusa tenggara , sulawesi, maluku, & papua	3.700	6.500	2.500	5.000

Dari seluruh rangkaian proses yang telah dikerjakan, dapat diketahui bahwa setiap jenis rute memiliki medan yang berbeda dan terkadang sebagian rute memiliki kesamaan medan dari perbedaan rute dan medan sehingga biaya atau tarif akan menyesuaikan. Dan pada penelitian ini akan menghasilkan rute angkutan roda dua dengan memperkirakan mana yang memiliki biaya paling sedikit dan nantinya *system* akan merekomendasikan angkutan roda dua dengan jarak yang paling dekat dengan pengguna seperti pada gambar 3.9.

Gambar 3.9 *Output* aplikasi

3.2.3 Perancangan Aplikasi

Pada tahapan ini akan dijelaskan mengenai perencanaan *system* yang akan dibangun dengan menggunakan pemodelan *UML* sehingga akan memudahkan pengoperasian dari perangkat tersebut. Pemodelan disini akan digambarkan ke dalam *Sequence Diagram*, *Use Case* serta *User Interface*.

A. Scenario Diagram

Nama *Use Case* : Menu Datar Sebagai Member/Ojek

Aktor : Penumpang

Deskripsi : Aktor menggunakan *menu* ini untuk mendaftar sebagai *member* baru atau ojek baru.

Tabel 3.5 *Scenario Diagram* menu daftar sebagai *member*/ojek

Aksi Aktor	Reaksi Sistem
Scenario Normal	
1. Menginputkan biodata sebagai <i>member</i> /ojek baru	
	2. Menyimpan data yang diinputkan
	3. Menampilkan menu utama

Nama *Use Case* : Menu Login

Aktor : Pengguna dan Ojek/Penyedia Jasa

Deskripsi : Aktor akan diberikan hak akses.

Tabel 3.6 *Scenario Diagram* menu login

Aksi Aktor	Reaksi Sistem
Scenario Normal	

1. Menginputkan biodata untuk mendapatkan hak akses masuk kedalam aplikasi	
	2. Melakukan pencocokan data dan memberikan hak akses
	3. Menampilkan <i>menu</i> utama

Nama *Use Case* : Menu Cari Tempat/Lokasi

Aktor : Penumpang

Deskripsi : Aktor menggunakan *menu* cari rute untuk mengetahui jenis medan rute dan angkutan roda dua terdekat.

Tabel 3.7 *Scenario Diagram menu cari tempat/lokasi*

Aksi Aktor	Reaksi Sistem
<i>Scenario Normal</i>	
4. Menginputkan nama tempat/lokasi	
	5. Masuk ke halaman <i>maps</i> dengan proses <i>autocomplete maps</i>
6. Memilih ojek yang terdekat	
	7. Menampilkan informasi data rute, medan, dan tarif

Nama *Use Case* : Menu Pilih Ojek

Aktor : Pengguna

Deskripsi : Aktor memilih ojek terdekat dengan posisi atau daerah atau lokasi awal dari pengguna.

Tabel 3.8 *Scenario Diagram* menu pilih ojek

Aksi Aktor	Reaksi Sistem
Scenario Normal	
1. Memilih menu pencarian	
	2. Menampilkan halaman <i>menu maps</i> dengan ojek disekitar posisi pengguna
3. Mengklik ojek yang terdekat	
	4. Menampilkan rute dari posisi <i>user</i> berada sebagai titik awal dan posisi tujuan

Nama Use Case : Menu Lihat Penumpang

Aktor : Ojek

Deskripsi : Aktor menjalankan fungsi ini untuk mengetahui rekomendasi penumpang yang akan dipilih.

Tabel 3.9 *Scenario Diagram* menu lihat penumpang

Aksi Aktor	Reaksi Sistem
Scenario Normal	
1. Memilih Menu Pantau Penumpang	
	2. Memperlihatkan penumpang yang siap dipilih
3. Mengklik penumpang yang ada dan terdekat	
	4. Menampilkan rute pada <i>maps</i> , dan melakukan <i>tracking</i> dari awal sampai tujuan.

Nama *Use Case* : Menu Terima Order

Aktor : Ojek

Deskripsi : Aktor menjalankan fungsi ini untuk mengambil penumpang yang akan menggunakan jasa.

Tabel 3.10 *Scenario Diagram* menu terima order

Aksi Aktor	Reaksi Sistem
<i>Scenario Normal</i>	
1. Mengklik penumpang yang ada dan terdekat	
	2. Menampilkan rute pada <i>maps</i> , dan melakukan <i>tracking</i> dari awal sampai tujuan.
	3. Menghitung jarak, medan dari rute yang dilalui dari awal mengantar sampai tempat tujuan

Nama *Use Case* : Menu Tentang Aplikasi

Aktor : User

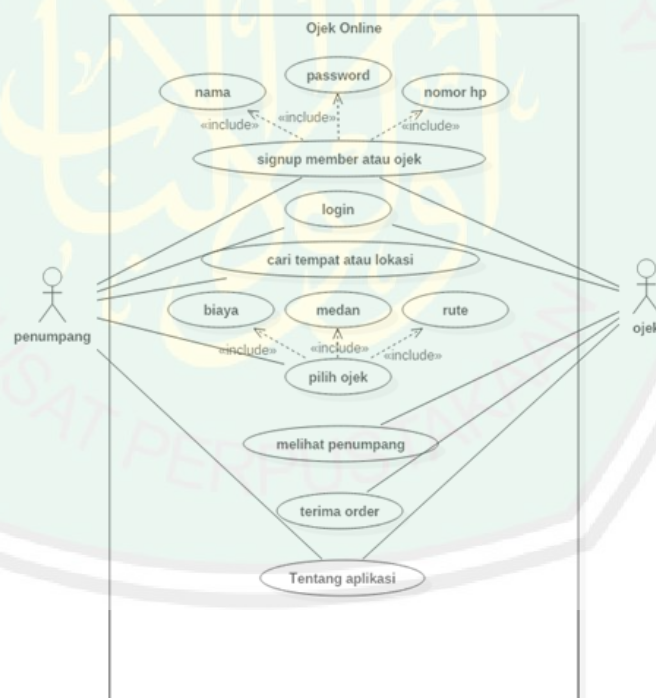
Deskripsi : aktor menjalankan fungsi ini untuk mengetahui tentang pengembang aplikasi ini.

Tabel 3.11 *Scenario Diagram* menu tentang aplikasi

Aksi Aktor	Reaksi Sistem
<i>Scenario Normal</i>	
1. Memilih menu Tentang Aplikasi	
	2. Masuk ke halaman Tentang Aplikasi
	3. Menampilkan informasi fitur dan terkait pengembangan aplikasi

B. Usecase Diagram

Semua perilaku yang dilakukan oleh *actor* akan dimodelkan sebagai *use case* yang mungkin dispesifikasikan dari realisasinya. *Use case* mendeskripsikan kumpulan urutan (*sequence*) dimana tiap urutan menjelaskan interaksi sistem dengan “sesuatu hal atau perilaku” di luar sistem (sering dinamakan dengan *actor*). *Use Case* menampilkan spesifikasi fungsional yang diharapkan dari sistem/perangkat lunak yang kelak akan dikembangkan. *Use Case* sangat penting dimanfaatkan untuk menangkap seluruh kebutuhan dan harapan pengguna. Berikut *usecase* dari pengguna yang digambarkan pada gambar 3.10.



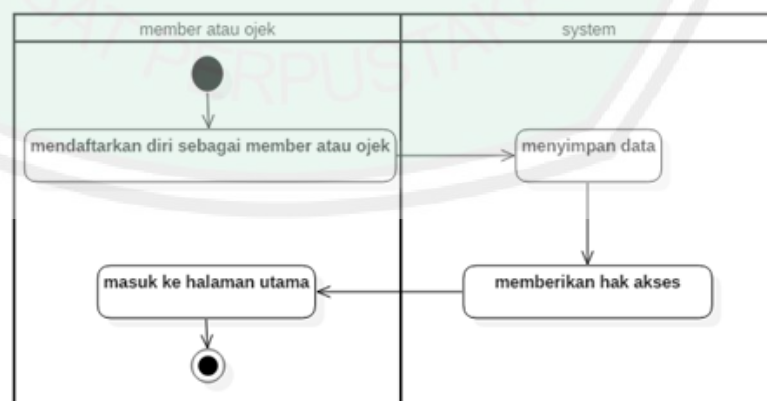
Gambar 3.10 Use Case Diagram

C. Activity Diagram

Diagram aktivitas atau *Activity Diagram* menggambarkan aliran fungsionalitas sistem. Pada aplikasi ini terdapat 7 (tujuh) *Activity Diagram*, yaitu sebagai seperti yang ada pada sub bahasan a sampai g.

a) *Activity Diagram Signup Member* atau Ojek

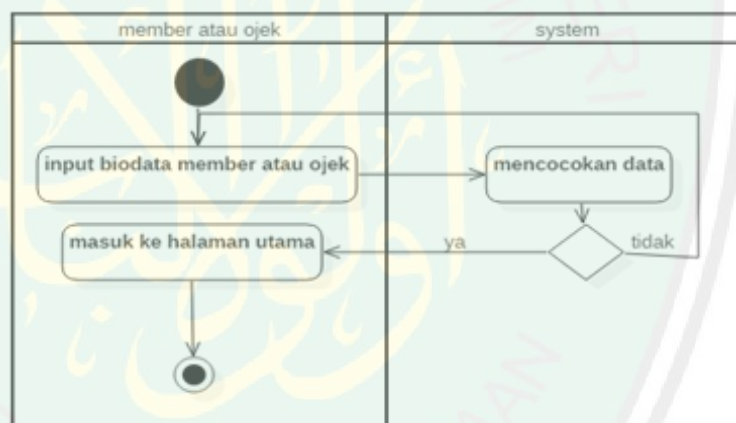
Activity diagram signup ini memodelkan aktifitas yang akan dilakukan oleh pengguna aplikasi ini baik sebagai *member* atau *ojek*. Dimana pengguna aplikasi ini harus mendaftar sebagai *member* terlebih dahulu untuk mendapatkan hak akses sehingga dapat menggunakan aplikasi. Dalam aktifitas pendaftaran secara otomatis sistem akan meminta pengguna untuk memasukkan biodata yang diperlukan oleh aplikasi untuk memberikan hak akses. Adapun biodata yang diperlukan untuk mendaftar adalah *user name*, *password*, dan nomor *handphone* yang aktif. *Activity Diagram signup member* atau *ojek* dimodelkan seperti pada gambar 3.11.



Gambar 3.11 *Activity Diagram signup member* atau *ojek*

b) Activity Diagram Login

Activity Diagram login dibuat untuk memodelkan aktifitas yang akan dilakukan oleh *member* atau ojek saat akan masuk kedalam aplikasi. Sebelum masuk kedalam aplikasi, pengguna harus menginputkan biodata yang dijadikan sebagai data akses pada aplikasi. Dimana data tersebut harus telah ada dalam *database* ketika akan dicocokkan. Dan data tersebut merupakan data yang telah diinputkan melalui proses *signup* atau pendaftaran. Berikut alur *Activity Diagram login* yang digambarkan pada Gambar 3.12.

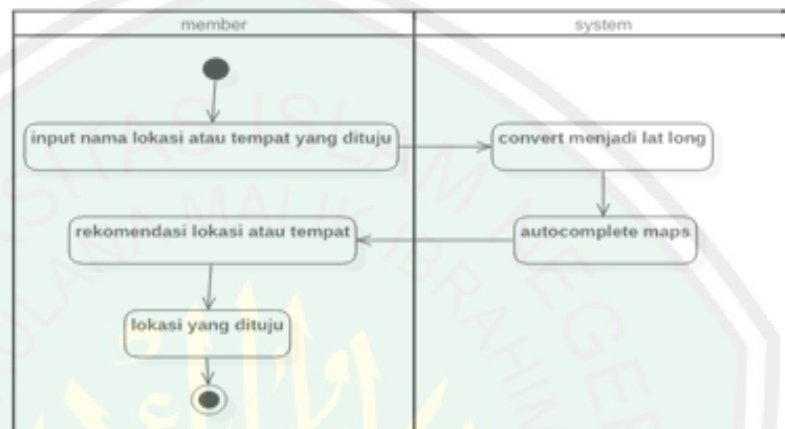


Gambar 3.12 Activity Diagram login

c) Activity Diagram Cari Tempat/Lokasi Tujuan

Activity Diagram cari tempat/lokasi tujuan dibuat untuk memodelkan aktifitas yang akan dilakukan oleh *member* atau penumpang. Penumpang yang akan mencari penyedia jasa atau ojek yang terdekat dari posisi dimana penumpang berada. Fitur ini akan memudahkan penumpang untuk memilih ojek atau berlangganan

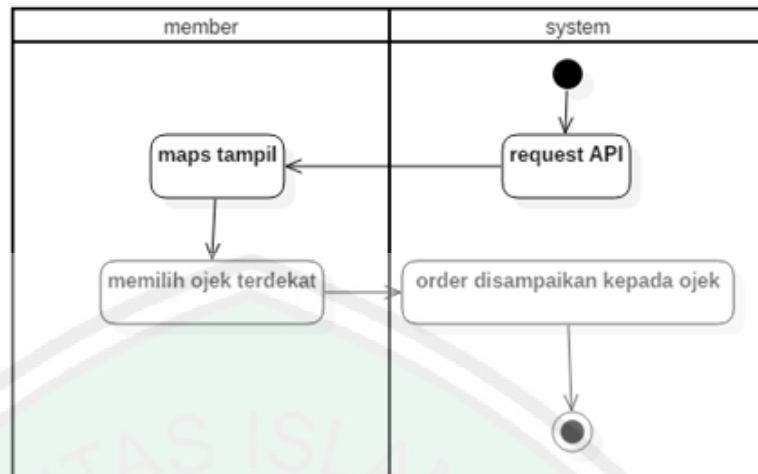
dengan penyedia jasa. Penumpang diberikan hak untuk memilih tanpa ada ketentuan yang berlaku selama *order* yang dilakukan sesuai dengan alur yang ada dalam sistem. *Activity Diagram* cari tempat/lokasi tujuan dimodelkan seperti pada gambar 3.13.



Gambar 3.13 Activity Diagram cari tempat/lokasi tujuan

d) Activity Diagram Pilih Ojek

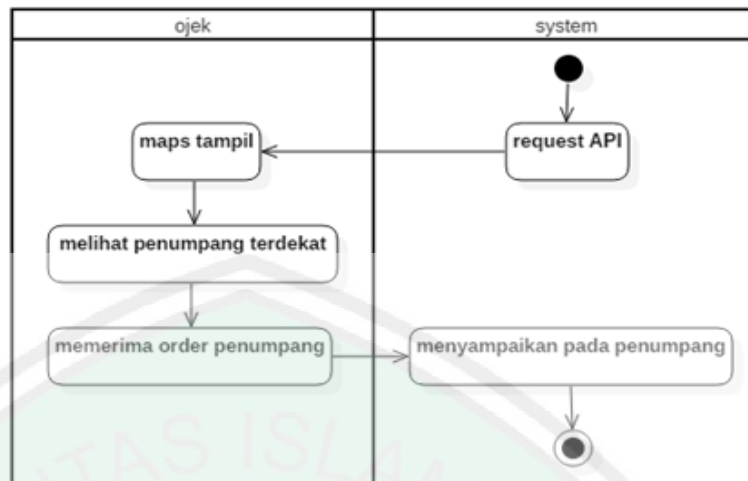
Activity Diagram pilih ojek dimulai dengan munculnya *maps* yang telah tampil ojek-okek yang terdaftar secara *Real Time*. Kemudian setelah *maps* tampil maka pengguna atau *member* dapat memilih satu ojek yang paling terdekat dari posisi pengguna. Pada saat memilih ojek, pengguna dapat menekan gambar ojek yang diinginkan dan jasa pengantaran atau penjemputan akan segera disorder oleh ojek yang sedang aktif. *Activity Diagram* pilih ojek disajikan dalam gambar 3.14.



Gambar 3.14 Activity Diagram pilih ojek

e) *Activity Diagram* Melihat Penumpang

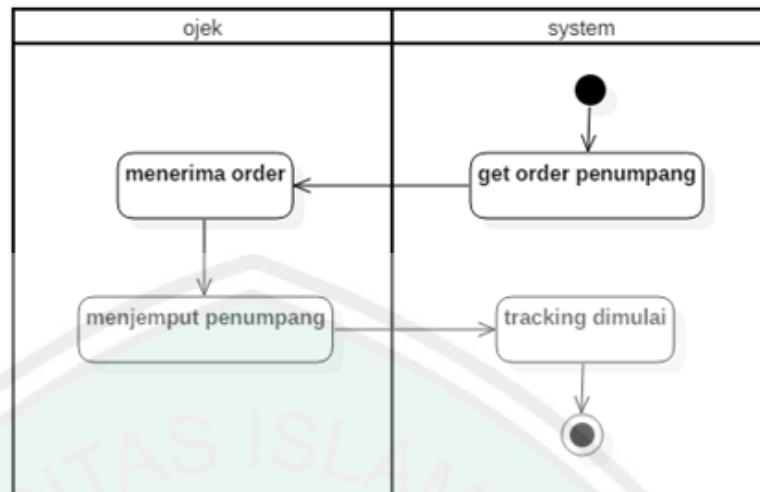
Pada *Activity Diagram* melihat penumpang dimodelkan tentang apa saja yang dilakukan ketika ojek akan melihat atau mencari *order*. Dimana pada awal aktivitas yang dilakukan adalah *maps* akan tampil kemudian ojek akan melihat penumpang yang sedang meminta jasa pengantaran atau penjemputan. Setelah mendapatkan *order* yang sesuai maka penumpang akan menunggu ojek datang menjemput ke lokasi penumpang. *Activity Diagram* melihat penumpang disajikan pada Gambar 3.15.



Gambar 3.15 Activity Diagram melihat penumpang

f) Activity Diagram Terima Order

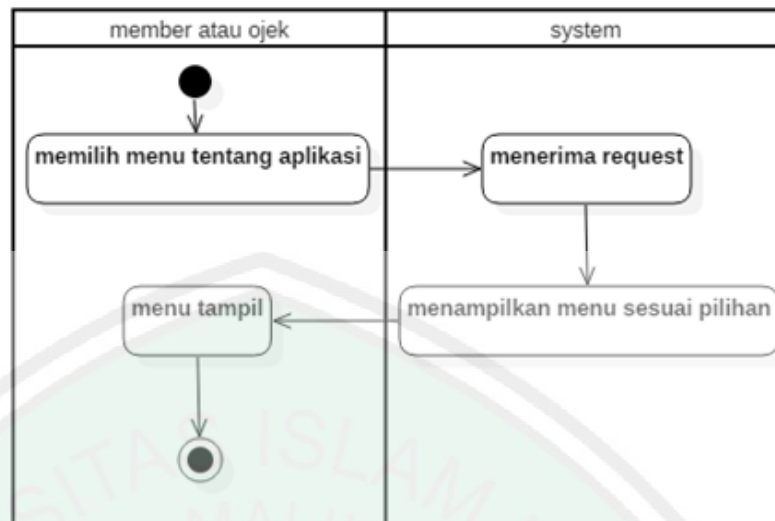
Pada Activity Diagram terima order dimodelkan tentang aktifitas yang dilakukan oleh ojek dalam menanggapi order dari penumpang yang ada. Dari order yang telah ditanggapi oleh ojek akan diberikan pemberitahuan kepada penumpang bahwa order telah diambil. Pada aktifitas ini diberikan fitur tambahan yaitu fitur *chat* untuk memudahkan komunikasi antara penumpang dan ojek. Sehingga terhindar dari kesalah pahaman pada transaksi. Activity diagram terima order disajikan pada Gambar 3.16.



Gambar 3.16 Activity Diagram terima order

g) Activity Diagram Tentang Aplikasi

Pada Activity Diagram tentang aplikasi memodelkan aktifitas yang akan dilakukan oleh pengguna aplikasi. Ketika pengguna mengalami kesulitan dalam memakai atau mengoperasikan aplikasi. Karena dirasa tuntunan penggunaan sangat penting dalam sebuah aplikasi maka ditambahkan fitur tentang aplikasi agar memudahkan pengguna. Activity Diagram melihat penumpang disajikan pada Gambar 3.17.



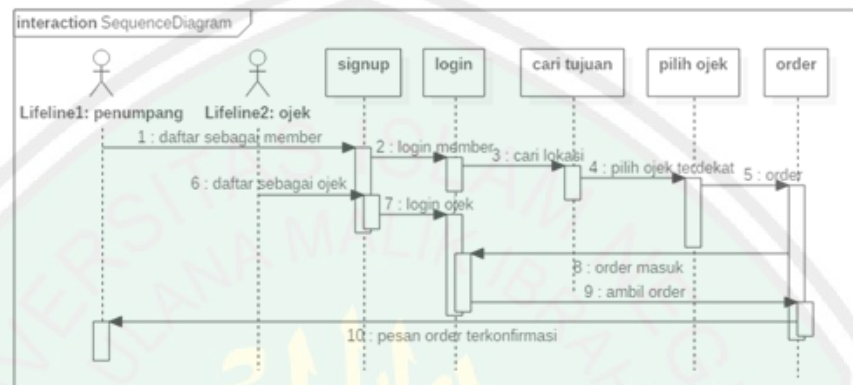
Gambar 3.17 Activity Diagram tentang aplikasi

D. Sequence Diagram

Diagram sekuensial atau *Sequence Diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Pada Aplikasi ini dijelaskan kedalam satu alur yang berurutan dari proses awal sampai dengan proses akhir. Adapun pada pemodelan *sequence* yang telah dibuat terdapat dua *line actor* yakni penumpang dan ojek dan pada komponen selanjutnya terdapat *life line* diantaranya *signup*, *login*, cari tujuan, pilih ojek, *order*. Alur pemodelan dapat diuraikan menjadi 3 yaitu alur yang dilakukan oleh penumpang, ojek dan proses yang terjadi pada system. Adapun uraiannya yakni pada proses awal pengguna atau *member* harus melakukan pendaftaran diri pada fitur *signup* agar dapat mendapatkan akses masuk kedalam aplikasi. Selanjutnya setelah terdaftar pengguna dialihkan untuk melakukan *login* agar dapat masuk kedalam *menu* aplikasi. Setelah itu pengguna akan masuk sesuai dengan akses masing-masing baik sebagai penumpang atau ojek. Kemudian untuk penumpang ada *menu* untuk

mencari lokasi tujuan, *order* ojek dan melihat ojek mana yang paling terdekat. Dan ojek dapat menerima *order* yang telah masuk untuk mengambil penumpang kemudian segera menjemput atau mengantarkan.

Untuk lebih jelasnya *Sequence Diagram* disajikan pada Gambar 3.18.

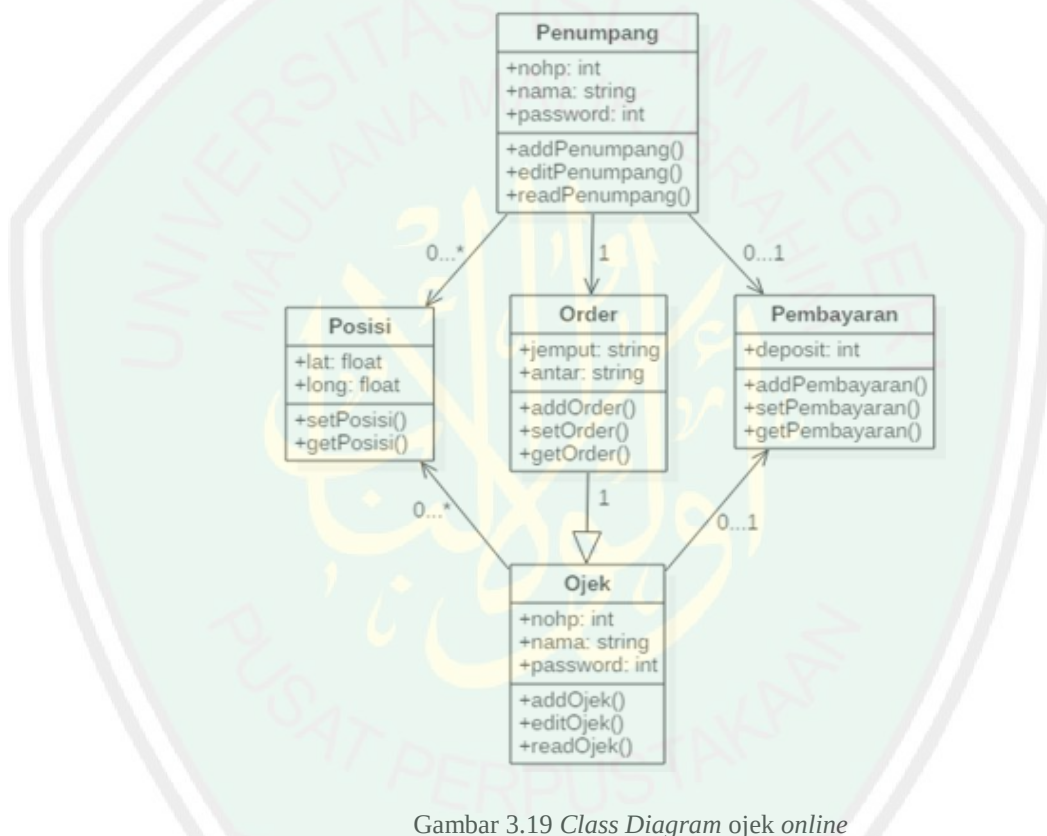


Gambar 3.18 *Sequence Diagram* ojek online

E. Class Diagram

Class Diagram merupakan implementasi bentuk dalam operasi dan relasi. Dari pembuatan *Class Diagram* pada aplikasi ini didapatkan terdapat beberapa bentuk class dan model relasi. Adapun beberapa bentuk class yang terdapat pada pemodelan adalah class penumpang, ojek, order, posisi, dan pembayaran. Kemudian untuk bentuk relasi yang digunakan menggunakan bentuk 1, 0...* dan 0...1 pada relasi antar *class* yang ada. Untuk uraian relasi dari pemodelan yang telah dibuat adalah pada hubungan antara *class* penumpang dengan *class* posisi menggunakan bentuk relasi 0...* yang berarti relasi dapat mengakses lebih dari beberapa atribut dari *class* yang beralasi begitupun untuk *class* ojek dengan *class* posisi. Kemudian untuk hubungan antara *class* penumpang atau *class* ojek dengan *class* order menggunakan bentuk relasi 1 yang berarti *class* yang berelasi hanya dapat

mengakses 1 atribut dari *class* yang berhubungan. Dan bentuk hubungan yang terakhir digunakan pada pemodelan ini adalah 0...1 yang terdapat pada hubungan *class* pembayaran dan *class* penumpang ataupun *class* ojek yang berarti *class* dapat mengakses *atribut* pada *class* yang berhubungan boleh 0 atau 1. ntuk lebih jelasnya *Class Diagram* disajikan pada Gambar 3.19.



Gambar 3.19 *Class Diagram* ojek online

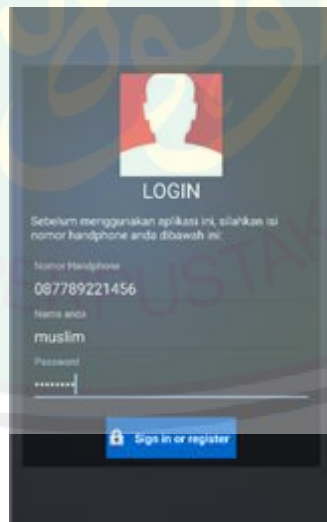
F. User Interface

User Interface adalah sebuah desain yang digunakan sebagai penghubung antara manusia sebagai pengguna dengan *software* pada perangkat sehingga akan memudahkan pengoperasian dari perangkat tersebut. Mudahnya, *UI* merupakan desain tampilan pada layar perangkat seperti yang dimiliki oleh *smartphone*, *tablet* dan berbagai jenis perangkat lainnya. Atau

sebagian orang awam lebih mengenalnya sebagai tema. Adapun *UI* yang digunakan dalam pembuatan Aplikasi ini pada *menu* di dalam aplikasi yang dibangun ini, terdapat beberapa *Menu* yaitu :

1. *Menu login*
2. *Menu utama*
3. *Menu pencarian tempat atau lokasi*

Pada *Menu Login* pengguna atau *member* akan diharuskan untuk masuk dengan data yang telah didaftarkan sebelumnya. Dari data inputan tersebut akan dicocokkan dengan data yang ada pada *database*. Kemudian bila data berhasil terverifikasi maka pengguna akan diberikan hak akses untuk masuk kedalam aplikasi. Berikut merupakan Gambar Rancangan *user interface menu login* pada aplikasi yang dibangun ini yang terdapat pada gambar 3.20.



Gambar 3.20 *Menu login*

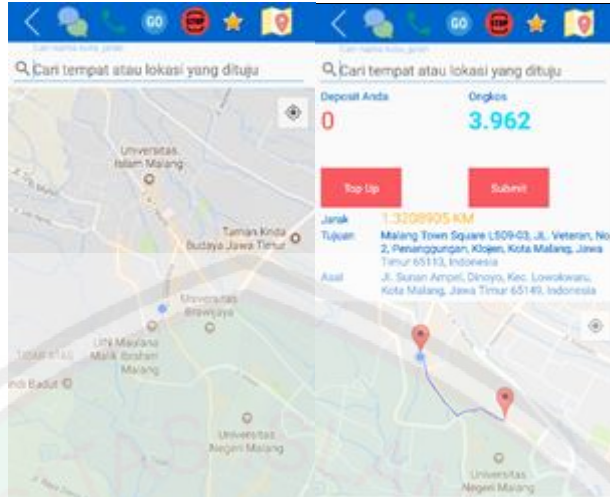
Selanjut akan masuk kedalam *menu utama* yang didalamnya terdapat beberapa *sub menu*. Adapun *sub menu* yang terdapat didalamnya antara lain

antar, jemput, *argo history*, dan *deposit*. Yang masing-masing dari *sub menu* akan merujuk pada *menu* lain yang memiliki fitur yang berbeda. Adapun tampilan dari *menu* utama digambarkan pada Gambar 3.21.



Gambar 3.21 Menu utama

Pada tampilan *menu* pencarian tempat atau lokasi, terdapat beberapa fitur seperti *chat*, *call*, *go/stop*, *rating*, dan *open Google Maps*. Dengan fungsi utama *menu* yaitu cari tempat atau lokasi, tampilan *maps* dan *detail order*. Kemudian penjelasan dari fitur yang ada antara lain *chat* digunakan untuk berkomunikasi antara penumpang dan ojek, *call* digunakan untuk melakukan pemanggilan kepada ojek, *go/stop* untuk memulai perjalanan, *rating* untuk memberikan peringkat layanan kepada ojek, dan *open maps* untuk melihat *detail maps*. digambarkan pada gambar 3.22.



Gambar 3.22 Menu pencarian tempat atau lokasi

3.3 Implementasi System

Pada tahap ini desain harus diimplementasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman Android dan *tools* pendukung lainnya dengan menyertakan penerapan kombinasi metode *Location Based Service* dan *Haversine*. Sehingga hasilnya berupa *Mobile App* untuk ojek online daerah NTB.

3.4 Pengujian System

Pada Tahapan uji coba yang akan dilakukan pada penelitian ini yaitu :

1. Verifikasi.

Tahap verifikasi ini adalah pengecekan kesesuaian hasil implementasi perangkat lunak dengan desain yang telah dirancang sebelumnya.

2. Pengujian akurasi metode

Pengujian keakuratan metode ini dilakukan untuk mengukur tingkat akurasi dari pengombinasian metode yang dipakai dalam penelitian ini, yakni *Location Based Service* dan *Haversine*. Pengujian akurasi metode ini dilakukan dengan membandingkan observasi yang dilakukan langsung, dengan hasil keluaran dari program yang telah dibangun.

Accuracy sendiri adalah prosentase dari total data uji coba yang benar diidentifikasi. Menurut (Gorunescu, 2011) standar tingkat akurasi dari hasil pengukuran adalah sebagai berikut:

- Akurasi 90% - 100% = *Excellent classification*
- Akurasi 80% - 90% = *Best classification*
- Akurasi 70% - 80% = *Fair classification*
- Akurasi 60% - 70% = *Poor classification*
- Akurasi 50% - 60% = *Failure*

Pada tahap *testing* metode dilakukan dengan memanfaatkan penggunaan perbandingan antara hasil dari metode *haversine* yang digunakan dengan hasil dari turunan hitungan *euclid*, serta jarak yang ada pada *google maps*. Adapun yang dibandingkan yaitu perkiraan jarak yang dihasilkan dan digunakan (*haversine*, *euclid*, dan *google maps*) serta perkiraan medan yang dihasilkan dengan menurunkan fungsi rumus *haversine* kemudian diambil korelasi sudut busurnya dan dimasukan kedalam *kontur kemiringan* untuk mengetahui medan. Dari proses perbandingan dapat dilihat akurasi jarak dan medan terbaik dihasilkan oleh metode yang digunakan. Dan dapat dilihat presentase akurasi dari metode

yang digunakan, kemudian dibandingkan dengan parameter akurasi masuk kedalam kategori yang mana.

Tabel 3.12 *testing* metode

testing metode haversine menggunakan euclid						
tempat	Latitude	longitude	haversine		kontur kemiringan	
kota bima	-8.4818619	118.7739739	24.49000197	km	1.238954975	persen derajat
kota sape	-8.5746799	118.9759335			Google earth	
					1.3	114 ft 2.43 mi
			euclid			
			24.74260074 km			
			google maps			
			25 km			
			persentase akurasi metode			
			jarak		medan	
			98.97909368		95.30422885	

3.5 Penarikan Kesimpulan Sementara

Penarikan kesimpulan dapat dilakukan setelah melalui tahapan perancangan atau pemodelan sistem, implementasi sistem, serta pengujian sistem telah selesai. Kemudian setelah hasil pengujian sistem telah didapatkan, maka selanjutnya dilakukan langkah evaluasi terhadap sistem yang telah dibangun berdasarkan hasil pengujian sebelumnya. Setelah melalui tahap evaluasi, maka tahap pengambilan kesimpulan dapat dilakukan.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Tahap implementasi pengembangan perangkat lunak merupakan tahap lanjutan dalam pengembangan perangkat lunak sehingga dapat digunakan dan sesuai dengan kebutuhan yang ada. Tahap ini merupakan lanjutan dari proses perancangan sistem, yaitu proses pemrograman perangkat lunak yang dilakukan sesuai dengan model dan desain sistem yang telah ada. Teknologi *platform* yang digunakan dalam pengembangan sistem ini adalah teknologi *platform* berbasis *mobile android*. Dengan diterapkannya konsep teknologi *platform* dari tahap penelitian sampai tahap implementasi dalam *mobile app online* menggunakan *location based service & metode haversine* dalam pengimplementasian jarak & medan.

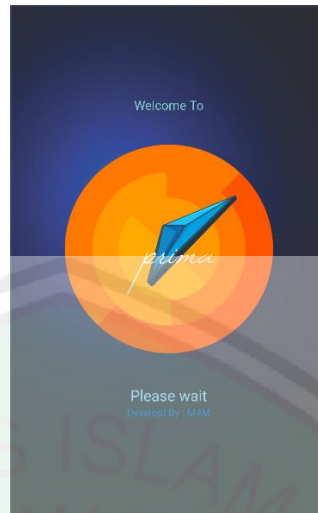
Tujuan dari implementasi sistem adalah untuk menerapkan perancangan yang telah dilakukan terhadap sistem, sehingga dapat menghasilkan perangkat lunak yang sesuai dengan kebutuhan. Pada proses pengimplementasian dilakukan secara runtun dimulai dari pengambilan latar belakang masalah sampai pada perancangan model sistem dan diimplementasikan kedalam perangkat lunak yang akan dibangun. Implementasi metode diterapkan pada sistem menggunakan perantara bahasa pemrograman *java* dan pada interaksi tampilan dengan pengguna menggunakan bahasa pemrograman *xml*. Dari konsep teknologi *platform mobile android* yang digunakan akan menghasilkan perangkat lunak yang dapat langsung dilihat bentuk implementasinya menggunakan *smartphone*, dengan *level* tahap percobaan implementasi yang bertingkat sesuai dengan panduan pengembangan yang telah distandarkan oleh *vendor platform*.

4.1 Implementasi *Interface*

Implementasi interface merupakan tampilan sistem yang berinteraksi dengan pengguna. Pada pengembangan tampilan interaksi sistem dibangun menggunakan bahasa pemrograman *xml* yang dapat dirancang dan dikembangkan sesuai dengan desain sistem yang telah dibuat. Adapun tampilan tidak akan dapat berinteraksi dengan pengguna tanpa diberikan *action code* menggunakan bahasa pemrograman *java* sehingga bahasa pemrograman yang digunakan akan berinteraksi satu sama lain. Penggunaan bahasa pemrograman *xml* sebagai perantara tampilan dikarenakan *xml* dapat dibuat dan dimanfaatkan kedalam bentuk *object* sehingga memudahkan penggunaan bahasa pemrograman *java* untuk mengalokasi *action* sesuai dengan *request object* pada setiap transisi yang terjadi pada *xml*.

4.1.1 Tampilan *Splash Screen*

Pada tampilan *splash screen*, tampilan *loading* awal saat membuka aplikasi. Durasi waktu yang digunakan dalam *flash screen* ini berdurasi 5 detik, sehingga ada jeda transisi ketika akan masuk pada menu utama dalam aplikasi. Dibuatnya *splash screen* dapat memperkenalkan pengembang aplikasi serta memberi sambutan pada pengguna. Ada beberapa penggunaan material dalam tampilan sehingga dapat menampilkan efek gambar bergerak. Tampilan *splash screen* terdapat pada gambar 4.1.



Gambar 4.1 Tampilan splash screen

Untuk mengontrol aksi transisi berpindahnya *layout* ke *layout* selanjutnya memanfaatkan pengimplementasian *source code* pada *activity* menggunakan java yang terdapat seperti pada gambar 4.2.

```

ActivitySplashScreen.java

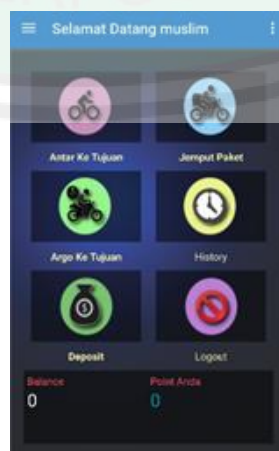
public class Splash extends Activity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash_activity);
        this.loading = (ImageView) findViewById(R.id.loading_img);
        animationDrawable = (AnimationDrawable)
            this.loading.getDrawable();
        Thread logoTimer = new Thread() {
            public void run() {
                try {
                    int logoTimer = 0;
                    while (logoTimer < 3000) {
                        sleep(100);
                        logoTimer = logoTimer + 100;
                    };
                    if (isLoggedIn) {
                        startActivity(new Intent("MainActivity"));
                    } else {
                        startActivity(new Intent("LoginActivity"));
                    }
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                finally {
                    finish();
                }
            }
        };
        logoTimer.start();
    }
}

```

Gambar 4.2 Source code activity splash screen

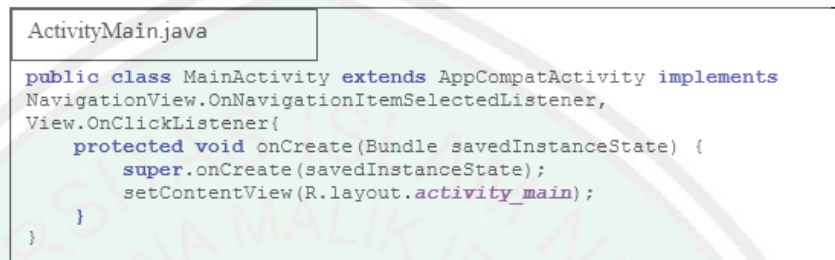
4.1.2 Menu Utama

Pada tampilan halaman utama aplikasi, akan terdapat beberapa *icon* aplikasi yang mempermudah *user* untuk menggunakan aplikasi. *Menu* yang pertama adalah menu , menu ini digambarkan dengan *icon* seorang pengendara sepeda motor yang di dalam menu tersebut berfungsi untuk melihat *maps* untuk menentukan tempat yang akan menjadi tempat tujuan. *Menu* yang kedua yakni menu jemput paket adalah menu untuk menjemput barang atau paket sesuai dengan permintaan pengguna. Terdapat menu tambahan lain seperti argo ke tujuan yang bertujuan memudahkan pengguna untuk melakukan permintaan jasa tanpa harus dijemput dari tempat asal melainkan langsung diantar menuju tempat yang dituju. Kemudian terdapat *menu history* yang memudahkan pengguna untuk melakukan kontrol terhadap pemakaian jasa antar jemput atau jasa lain, sehingga penggunaan jasa dapat teratur. Dan *menu* deposit menjadi *menu* yang wajib dipenuhi oleh penumpang dimana penumpang harus mengisi *virtual money* sebelum menggunakan jasa seperti pada gambar 4.3.



Gambar 4.3 *Menu utama*

Untuk mengontrol setiap aksi tombol-tombol yang ada pada *layout* menggunakan pengimplementasian *source code* pada *activity* menggunakan java yang terdapat seperti pada gambar 4.4.



```

ActivityMain.java

public class MainActivity extends AppCompatActivity implements
NavController.OnNavigationItemSelectedListener,
View.OnClickListener{
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

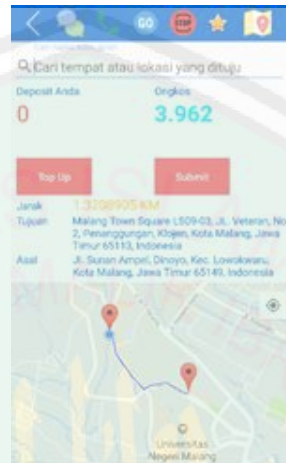
```

Gambar 4.4 *Source code activity main*

4.1.3 Menu Rute Maps

Pada tampilan *menu rute maps*, berfungsi untuk memudahkan pengguna untuk menentukan tempat tujuan dan melihat rute mana yang akan dilewati serta memilih penyedia jasa mana yang terdekat dengan posisinya saat melakukan pemesanan. Pada tampilan diberikan beberapa fungsi tambahan yang diharap dapat memudahkan pengguna dalam mengoperasikan aplikasi. Fungsi tambahan tersebut antara lain penambahan fitur *chat* yang dapat digunakan sebagai sarana komunikasi antara *driver* dan penumpang sehingga proses transaksi dapat berjalan sesuai yang diharapkan, fitur telepon yang dapat digunakan sebagai sarana komunikasi alternatif antara *driver* dan penumpang, fitur memulai *track* dan berhenti sebagai pengatur aktifitas pengantaran penumpang mulai dari awal menerima pemakaian jasa antar sampai dengan berakhir pada tujuan yang dituju, fitur memberikan *rating* pada *driver* diberikan kepada penumpang sebagai fitur untuk memberikan penilaian pada

driver, kemudian pengguna dapat memulai ulang *maps* dengan menekan *icon maps* sehingga *maps* akan mengosongkan riwayat perjalanan lama seperti pada gambar 4.5.



Gambar 4.5 Menu rute maps

Untuk menampilkan setiap aksi yang ada pada *maps* memanfaatkan beberapa *library* yang disediakan oleh *google* sehingga dapat diimplementasikan kedalam *source code* pada *activity* menggunakan *java* yang terdapat seperti pada gambar 4.6.

```

ActivityMaps.java

public class MapsActivity extends AbstractMapActivity implements
    OnMapReadyCallback, GoogleMap.OnInfoWindowClickListener,
    LocationListener, GoogleApiClient.OnConnectionFailedListener,
    GoogleApiClient.ConnectionCallbacks, GoogleMap.OnMapClickListener{
    public void onAttachFragment(Fragment fragment) {
        super.onAttachFragment(fragment);
    }
    public void onMapClick(LatLng latLng) {
        map.clear();
        map.addMarker(new MarkerOptions()
            .position(new LatLng(latLng.latitude,
            latLng.longitude))
            .title("Starting Point")

            .icon(BitmapDescriptorFactory.fromResource(R.drawable.antar))
        );
        myLatLng = latLng;
    }
    //LBS & Haversine
    private double getRadius(int inKm) {
        double latDistance = Math.toRadians(myLatLng.latitude -
        inKm);
        double lngDistance = Math.toRadians(myLatLng.longitude -
        inKm);
        double a = (Math.sin(latDistance / 2) *
        Math.sin(latDistance / 2)) +
        (Math.cos(Math.toRadians(myLatLng.latitude))) *
        (Math.cos(Math.toRadians(inKm))) *
        (Math.sin(lngDistance / 2)) *
        (Math.sin(lngDistance / 2));
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
        double dist = 6371 / c;
        if (dist < 50) {
        }
        return dist;
    }

    public void DrawRoute(LatLng from, LatLng to) {
        //kontur kemiringan
        mJarakVal = mDistance;
        mJarak.setText("" + mDistance + " KM");

        mhd1 = (180 * myLatLng.latitude - 180 * myLatLng.longitude);
        mhd2 = (180 * mToLat - 180 * mToLng);
        mhd = mhd1 - mhd2 * 100000 / 1;
        mMedan.setText("" + mhd + "");
        int o = (int) (mDistance * mTarif); //ongkos 1000 tiap
        kilometer
        DecimalFormat df = new DecimalFormat("###,###.##"); // or
        pattern "###,###.###"
        mOngkos.setText(df.format(o));
        mOngkosVal = o;
    }
}

```

Gambar 4.6 Source code activity maps

4.1.4 Menu History

Di dalam *menu history*, terdapat informasi penggunaan dan sisa dari deposit dari pengguna. Didalam *menu history* terdapat informasi lain tentang tanggal terakhir penggunaan jasa, jarak yang ditempuh serta biaya

yang dikeluarkan. Pada *menu history* ini lebih seperti rekapan penggunaan jasa oleh pengguna sehingga pengguna dapat memonitoring dan mengelola deposit penggunaan jasa dengan baik seperti pada gambar 4.7.



Gambar 4.7 Menu history

Source code untuk menampilkan setiap riwayat transaksi seperti pada gambar 4.8.

ActivityHistory.java

```
public class DompActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.domp_activity);
        txtSaldo=(TextView) findViewById(R.id.txtSaldo);
        lstTrans=(ListView) findViewById(R.id.lstTrans);

        mSetting = getSharedPreferences("setting_gojek",
            Context.MODE_PRIVATE);
        mName=mSetting.getString("nama", "Guest");
        mNomorHp=mSetting.getString("no_hp", "0000000000");
        int mDeposit=new Deposit(this).Saldo(mNomorHp);
        DecimalFormat df = new DecimalFormat("###,###.##");
        txtSaldo.setText(df.format(mDeposit));
        TransactionAdapter transactionAdapter=new
        TransactionAdapter(this,mNomorHp);
        lstTrans.setAdapter(transactionAdapter);
    }
}
```

Gambar 4.8 Source code activity history

4.1.5 Menu Deposit

Pada *menu deposit* ditunjukkan untuk memudahkan pengguna untuk menambahkan saldo pada *virtual money* yang disediakan. Pengguna diberikan tuntunan untuk mengisi data terlebih dahulu sebelum melakukan transaksi pengisian deposit saldo. Adapun yang wajib diisikan oleh pengguna adalah nomer rekening *bank* yang digunakan oleh pengguna, nama pemilik rekening, serta jumlah nominal yang akan diisi kedalam deposit seperti pada gambar 4.9.

Gambar 4.9 Menu deposit

Untuk menampilkan dan menyimpan data deposit pada setiap transaksi yang dilakukan oleh pengguna pada *layout* memanfaatkan beberapa data dari *API* yang telah ada sehingga dapat diimplementasikan kedalam *source code* pada *activity* menggunakan java yang terdapat seperti pada gambar 4.10.


```

ActivityDeposit.java

public class DepositActivity extends AppCompatActivity implements
View.OnClickListener {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.deposit_activity);
    }
    public void onClick(View view) {
        int id=view.getId();
        if(id==R.id.submit){
            Deposit deposit=new Deposit(this);
            if(
deposit.Save(mNoHp,mBank.getText().toString(),mNama.getText().
toString(),

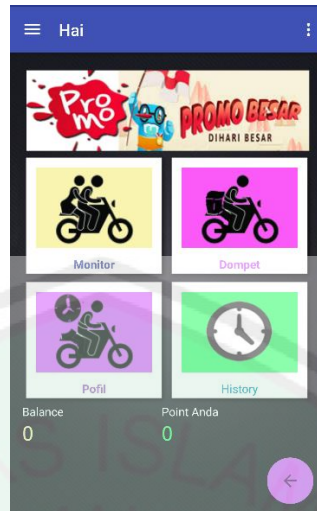
Integer.parseInt(mJumlah.getText().toString())) ) {
                String msg="Sukses data sudah masuk, tunggu
beberapa saat untuk verifikasi dan masuk ke saldo deposit
anda.";
                Toast.makeText(this,msg,Toast.LENGTH_LONG).show();
                finish();
            } else {
                Toast.makeText(this,deposit.getError(),Toast.LENGTH_LONG).show
                ();
            }
        }
    }
}

```

Gambar 4.10 Source code activity deposit

4.1.6 Menu Driver

Driver dapat melakukan akses terhadap beberapa fitur yang telah disediakan. Fitur yang tersedia untuk *driver* sedikit berbeda dengan yang ada pada pengguna biasa. Adapun fitur tersebut antara lain yaitu fitur monitoring untuk memonitor penumpang, *history* untuk melihat riwayat aktifitas transaksi jasa yang telah dilakukan, serta fitur profil *driver* dan dompet sebagai fitur rekap pendapatan *driver* yang akan terisi sesuai dengan jumlah yang didapat dari tarif pengantaran penumpang ataupun transaksi jasa lainnya yang tersedia pada fitur. Adapun tampilan fitur-fitur yang tersedia untuk *driver* dapat dilihat seperti pada gambar 4.11.



Gambar 4.11 Tampilan menu drive

Untuk mengontrol setiap aksi tombol yang ada pada *layout* menggunakan pengimplementasian *source code* pada *activity* menggunakan java yang dapat dilihat seperti pada gambar 4.12.

ActivityMainDriver.java

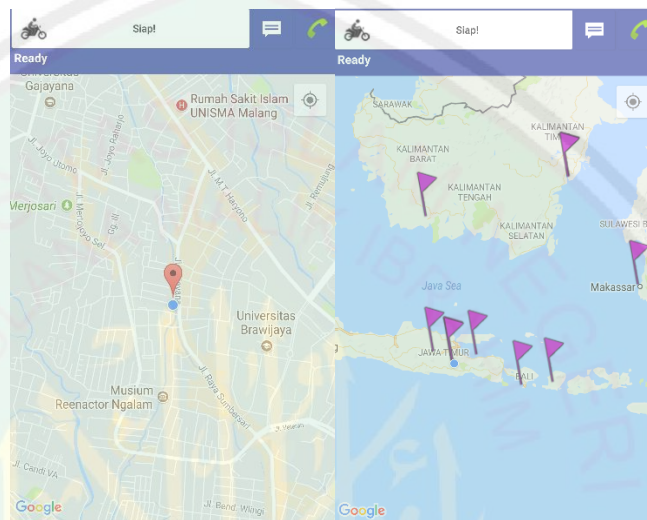
```
public class MainActivity extends AppCompatActivity implements
    NavigationView.OnNavigationItemSelectedListener,
    View.OnClickListener {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Gambar 4.12 Source code activity main driver

4.1.7 Menu Monitoring Penumpang

Pada *menu monitoring* ditujukan untuk memudahkan *driver* untuk melihat penumpang yang sedang aktif disekitar lokasi tersebut. *Driver* dapat memperkirakan penumpang mana yang akan diorder dengan cara melihat pos-pos yang telah digambarkan dengan tanda bendera-bendera pos yang terlihat pada *maps*. Setelah *driver* menerima *order* dari penumpang maka *driver* dapat menjemput penumpang dan

mengantarkannya ketempat tujuan penumpang. Pada saat aktifitas pengantaran penumpang *driver* dapat melakukan *tracking* pada rute yang akan dilewatinya. Adapun tampilan dari menu dapat dilihat seperti pada gambar 4.13.



Gambar 4.13 Menu pencerian penumpang

Untuk menampilkan setiap aksi yang ada pada *maps* memanfaatkan beberapa *library* yang disediakan oleh *google* sehingga dapat diimplementasikan kedalam *source code* pada *activity* menggunakan *java*. Pemberian *tag* pos-pos pada *maps* yang digambarkan dengan *icon* bendera mengambil data dari *order* yang diberikan oleh penumpang sehingga *driver* dapat memantau *order* penumpang mana yang paling dekat dengan posisi *driver* saat melakukan *monitoring order* seperti pada gambar 4.14.

```

ActivityMaps.java

public class MapsActivity extends AbstractMapActivity implements
    OnMapReadyCallback, GoogleMap.OnInfoWindowClickListener,
    LocationListener, GoogleApiClient.OnConnectionFailedListener,
    GoogleApiClient.ConnectionCallbacks, GoogleMap.OnMapClickListener{
    public void onAttachFragment(Fragment fragment) {
        super.onAttachFragment(fragment);
    }
    public void onMapClick(LatLng latLng) {
        map.clear();
        map.addMarker(new MarkerOptions()
            .position(new LatLng(latLng.latitude,
            latLng.longitude))
            .title("Starting Point")

            .icon(BitmapDescriptorFactory.fromResource(R.drawable.antar))
        );
        myLatLng = latLng;
    }
    //LBS & Haversine
    private double getRadius(int inKm) {
        double latDistance = Math.toRadians(myLatLng.latitude -
        inKm);
        double lngDistance = Math.toRadians(myLatLng.longitude -
        inKm);
        double a = (Math.sin(latDistance / 2) *
        Math.sin(latDistance / 2)) +
        (Math.cos(Math.toRadians(myLatLng.latitude)) *
        (Math.cos(Math.toRadians(inKm))) *
        (Math.sin(lngDistance / 2)) *
        (Math.sin(lngDistance / 2)));
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
        double dist = 6371 / c;
        if (dist < 50) {
        }
        return dist;
    }
}

```

Gambar 4.14 Source code activity main driver

4.2 Pengujian Sistem

Setelah tahap implementasi selesai dan terpenuhi maka dilakukan pengujian sistem agar aplikasi yang dibuat sesuai dengan perancangan, serta layak untuk digunakan oleh pengguna pada keadaan yang sebenarnya. Pengujian pada sistem ini adalah dengan membandingkan data rute baik dalam bentuk jarak ataupun dalam bentuk medan serta tujuannya yang ditentukan secara *manual*, dengan data rute yang dijadikan *sample* serta tujuannya yang diproses dengan menggunakan sistem yang telah dibangun.

4.2.1 Persiapan Data

Pada tahap persiapan ini data yang digunakan adalah berupa data *sample* lokasi baik yang lokasi awal maupun lokasi yang dituju. Data tersebut kemudian dipecah kedalam 2 kelompok berdasarkan *latitude* dan *longitude*. Data yang akan digunakan pada saat proses *testing* dapat dilihat pada tabel 4.1.

Tabel 4.1 Data *testing*

No.	sample location		
	Lokasi	latitude	longitude
1	<i>UIN Maliki</i>	-7.9476535	112.5955092
2	<i>Matos</i>	-7.9571941	112.6184881
3	<i>UnBraw</i>	-7.9476535	112.5955092
4	<i>UNM</i>	-7.9627744	112.6184781
5	<i>UMM</i>	-7.95763934	112.5955092
6	<i>ITN</i>	-7.9476535	112.6139389
7	<i>Unisma</i>	-7.9365167	112.6071692
8	<i>Poltek Malang</i>	-7.9465289	112.6155368
9	<i>Museum Brawijaya</i>	-7.9472072	112.6139054
10	<i>Balai kota Malang</i>	-7.9472072	112.6139054
11	<i>Ramayana mall</i>	-7.9472072	112.5955092
12	<i>Alun-alun malang</i>	-7.9476535	112.6139054
13	<i>Cyber Mall</i>	-7.9472072	112.6139054
14	<i>MAN 3 Malang</i>	-7.9472072	112.6139054
15	<i>RS Unisma</i>	-7.9472072	112.6139054
16	<i>Sardo</i>	-7.9472072	112.6139054
17	<i>Mc D</i>	-7.9472072	112.6139054
18	<i>Bandara Asrikaton</i>	-7.9472072	112.6139054
19	<i>Stasiun Pujasera</i>	-7.969457	112.5955092
20	<i>Terminal Arjosari</i>	-7.9476535	112.5773415
21	<i>Malang Strudel</i>	-7.969457	112.5773415
22	<i>Mie Setan</i>	-7.969457	112.5773415
23	<i>RM Nelongso</i>	-7.969457	112.5773415
24	<i>Bakso Presiden</i>	-7.969457	112.5773415
25	<i>Jodipan</i>	-7.98835674	112.639978
26	<i>Terminal Mie</i>	-7.969457	112.5773415
27	<i>Martabak Agung</i>	-7.969457	112.5773415

28	Burger Buto	-7.969457	112.5773415
29	Mie Jogging	-7.969457	112.5773415
30	Bakpao Boldy	-7.969457	112.5773415
31	Sambal Mak Tin	-7.969457	112.5773415
32	Ketan Bubuk	-7.969457	112.5773415
33	Rujak Ndower	-7.969457	112.5773415
34	Warung Soto Sawah	-7.969457	112.5955092
35	Angkringan Podjok	-7.9476535	112.5773415
36	Jank-Jank wings	-7.969457	112.5773415
37	Banana Kingkong	-7.969457	112.5773415

4.2.2 Pengujian

Penggunaan data perbandingan pada proses *testing* oleh varian dari data. Proses *testing* menggunakan cara perbandingan yang ada pada *one-way anova* membuat data *sample* akan dibandingkan berdasarkan ukuran atau varian data dan akan menghasilkan data selisih dari varian data *sample* (Mahapoonyanont, 2010). Pada proses *testing* yang telah dilakukan digunakan perbandingan *one-way anova*. Dimana proses *testing* dilakukan pada waktu yang bersamaan yaitu pada waktu jam kerja kantor yang diberlakukan oleh Pemkot Malang di wilayah Malang yang mencakup waktu pagi pukul 08.00 wib hingga petang pukul 16.00 wib termasuk pada jam istirahat siang pukul 11.30 – 12.30 wib di wilayah lokasi awal dan lokasi tujuan yang sama. Adapun data yang dicari selisihnya adalah selisih harga, dan jarak sehingga dapat dilihat akurasi serta efektifitas dari implementasi yang telah dilakukan. Proses *testing* aplikasi menggunakan *one-way anova* dapat dilihat pada tabel 4.2.

Tabel 4.2 Proses *testing*

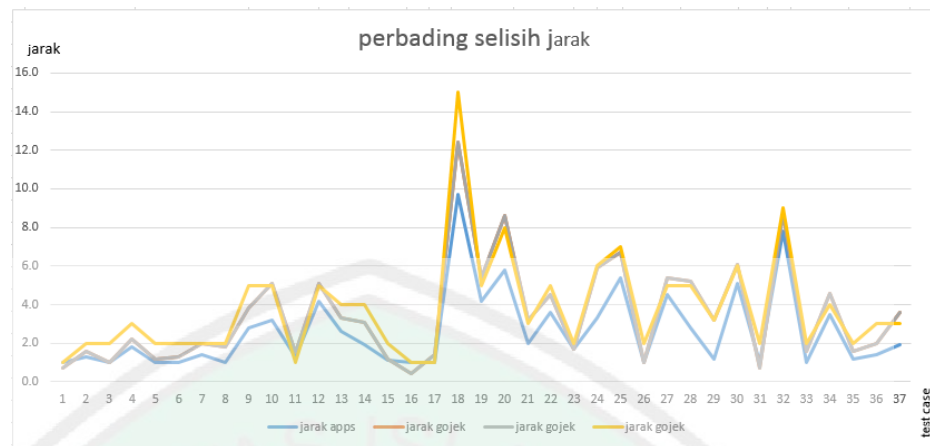
No.	apps				gojek				grab				golek			
	jarak apps			tarif apps	jarak gojek			tarif gojek	jarak gojek			tarif grab	jarak gojek			tarif golek
1	1.0	km	Rp.	2.500	0.7	km	Rp.	4.000	0.7	km	Rp.	4.000	1.0	km	Rp.	5.000
2	1.3	km	Rp.	3.380	1.6	km	Rp.	4.000	1.6	km	Rp.	4.000	2.0	km	Rp.	5.000
3	1.0	km	Rp.	2.500	1.0	km	Rp.	4.000	1.0	km	Rp.	4.000	2.0	km	Rp.	5.000
4	1.8	km	Rp.	4.438	2.2	km	Rp.	4.000	2.2	km	Rp.	5.000	3.0	km	Rp.	5.000
5	1.0	km	Rp.	2.500	1.2	km	Rp.	4.000	1.2	km	Rp.	4.000	2.0	km	Rp.	5.000
6	1.0	km	Rp.	2.500	1.3	km	Rp.	4.000	1.3	km	Rp.	4.000	2.0	km	Rp.	5.000
7	1.4	km	Rp.	3.723	2.0	km	Rp.	4.000	2.0	km	Rp.	4.000	2.0	km	Rp.	5.000
8	1.0	km	Rp.	2.500	1.8	km	Rp.	4.000	1.8	km	Rp.	6.000	2.0	km	Rp.	5.000
9	2.8	km	Rp.	7.201	3.8	km	Rp.	7.000	3.8	km	Rp.	7.000	5.0	km	Rp.	8.000
10	3.2	km	Rp.	8.207	5.1	km	Rp.	10.000	5.1	km	Rp.	8.000	5.0	km	Rp.	8.000
11	1.3	km	Rp.	3.336	1.5	km	Rp.	4.000	1.5	km	Rp.	4.000	1.0	km	Rp.	5.000
12	4.2	km	Rp.	10.639	5.1	km	Rp.	10.000	5.1	km	Rp.	8.000	5.0	km	Rp.	8.000
13	2.6	km	Rp.	6.584	3.3	km	Rp.	6.000	3.3	km	Rp.	6.000	4.0	km	Rp.	6.500
14	1.9	km	Rp.	4.815	3.1	km	Rp.	6.000	3.1	km	Rp.	4.000	4.0	km	Rp.	6.500
15	1.1	km	Rp.	2.771	1.2	km	Rp.	4.000	1.2	km	Rp.	4.000	2.0	km	Rp.	5.000
16	1.0	km	Rp.	2.500	0.4	km	Rp.	4.000	0.4	km	Rp.	4.000	1.0	km	Rp.	5.000
17	1.0	km	Rp.	2.500	1.4	km	Rp.	4.000	1.4	km	Rp.	4.000	1.0	km	Rp.	5.000
18	9.7	km	Rp.	24.438	12.4	km	Rp.	23.000	12.4	km	Rp.	19.000	15.0	km	Rp.	23.000
19	4.2	km	Rp.	10.554	5.4	km	Rp.	10.000	5.4	km	Rp.	9.000	5.0	km	Rp.	8.000
20	5.8	km	Rp.	14.567	8.6	km	Rp.	16.000	8.6	km	Rp.	14.000	8.0	km	Rp.	12.000
21	2.0	km	Rp.	5.039	3.2	km	Rp.	6.000	3.2	km	Rp.	5.000	3.0	km	Rp.	5.000
22	3.6	km	Rp.	9.226	4.5	km	Rp.	9.000	4.5	km	Rp.	8.000	5.0	km	Rp.	8.000
23	1.7	km	Rp.	6.991	1.7	km	Rp.	4.000	1.7	km	Rp.	7.000	2.0	km	Rp.	5.000
24	3.3	km	Rp.	8.447	5.9	km	Rp.	11.000	5.9	km	Rp.	9.000	6.0	km	Rp.	9.500
25	5.4	km	Rp.	12.561	6.7	km	Rp.	13.000	6.7	km	Rp.	10.000	7.0	km	Rp.	11.000
26	1.0	km	Rp.	2.500	1.0	km	Rp.	4.000	1.0	km	Rp.	4.000	2.0	km	Rp.	5.000
27	4.5	km	Rp.	11.236	5.4	km	Rp.	10.000	5.4	km	Rp.	9.000	5.0	km	Rp.	8.000
28	2.8	km	Rp.	7.065	5.2	km	Rp.	10.000	5.2	km	Rp.	10.000	5.0	km	Rp.	8.000
29	1.2	km	Rp.	5.086	3.2	km	Rp.	6.000	3.2	km	Rp.	5.000	3.2	km	Rp.	6.000
30	5.1	km	Rp.	9.140	6.1	km	Rp.	10.000	6.1	km	Rp.	10.000	6.0	km	Rp.	9.500
31	1.0	km	Rp.	2.500	0.7	km	Rp.	4.000	0.7	km	Rp.	4.000	2.0	km	Rp.	5.000
32	7.8	km	Rp.	19.707	8.7	km	Rp.	16.000	8.7	km	Rp.	14.000	9.0	km	Rp.	14.000
33	1.0	km	Rp.	2.723	1.6	km	Rp.	4.000	1.6	km	Rp.	4.000	2.0	km	Rp.	5.000
34	3.5	km	Rp.	8.838	4.6	km	Rp.	9.000	4.6	km	Rp.	7.000	4.0	km	Rp.	6.500
35	1.2	km	Rp.	3.127	1.6	km	Rp.	4.000	1.6	km	Rp.	4.000	2.0	km	Rp.	5.000
36	1.4	km	Rp.	3.527	2.0	km	Rp.	4.000	2.0	km	Rp.	5.000	3.0	km	Rp.	5.000
37	1.9	km	Rp.	4.939	3.6	km	Rp.	7.000	3.6	km	Rp.	6.000	3.0	km	Rp.	5.000

Tabel 4.3 Hasil *testing*

No.	selisih		selisih	
	km		harga	
1	0.85	km	Rp.	3.875
2	1.625	km	Rp.	4.095
3	1.25	km	Rp.	3.875
4	2.3	km	Rp.	4.610
5	1.35	km	Rp.	3.875
6	1.4	km	Rp.	3.875
7	1.85	km	Rp.	4.181
8	1.65	km	Rp.	4.375
9	3.85	km	Rp.	7.300
10	4.6	km	Rp.	8.552
11	1.325	km	Rp.	4.084
12	4.85	km	Rp.	9.160
13	3.3	km	Rp.	6.271
14	3.025	km	Rp.	5.329
15	1.375	km	Rp.	3.943
16	0.7	km	Rp.	3.875
17	1.2	km	Rp.	3.875
18	12.375	km	Rp.	22.360
19	5	km	Rp.	9.389
20	7.75	km	Rp.	14.142
21	2.85	km	Rp.	5.260
22	4.4	km	Rp.	8.557
23	1.775	km	Rp.	5.748
24	5.275	km	Rp.	9.487
25	6.45	km	Rp.	11.640
26	1.25	km	Rp.	3.875
27	5.075	km	Rp.	9.559
28	4.55	km	Rp.	8.766
29	2.7	km	Rp.	5.522
30	5.825	km	Rp.	9.660
31	1.1	km	Rp.	3.875
32	8.55	km	Rp.	15.927
33	1.55	km	Rp.	3.931
34	4.175	km	Rp.	7.835
35	1.6	km	Rp.	4.032
36	2.1	km	Rp.	4.382
37	3.025	km	Rp.	5.735

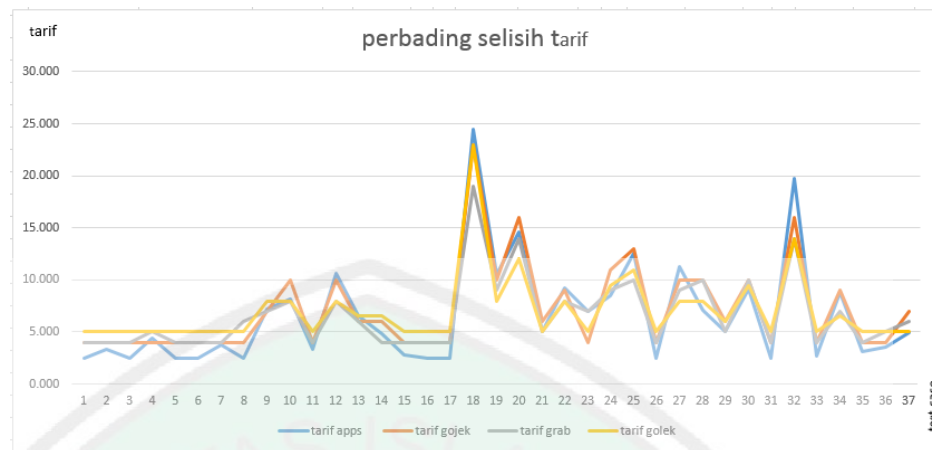
Proses *testing* dilakukan secara terurut berdasarkan penomoran data *sample*. Data yang digunakan adalah *random area* tidak berdasarkan jarak, rute atau parameter lain. Proses *testing* mengambil data *latitude* dan *longitude* pada data *sample* yang ada sesuai dengan penomoran yang diberikan sehingga proses *testing* terurut dan hasil dapat dilihat secara runtun. Data perbandingan yang digunakan diambil dari salah satu aplikasi yang telah ada sebelumnya dan telah banyak digunakan oleh pengguna yaitu gojek sehingga proses perbandingan dirasa akan mampu menghasilkan data yang dapat memperbaiki akurasi dan efektifitas dari aplikasi yang dibangun.

Dari table 4.2 dapat dijadikan kedalam bentuk grafik pelana yang didalamnya dapat dipresentasikan data jarak maupun harga, sehingga dapat dilihat secara detail berapa selisih baik jarak maupun harga dari perbandingan telah dilakukan. Adapun grafik yang dibuat akan mempresentasikan jarak dari aplikasi yang dibuat dengan jarak dari gojek. Dan akan membandingkan tarif dari aplikasi yang telah dibuat dengan tarif dari gojek. Sehingga akan dapat dilihat selisih perbandingan baik jarak maupun harga secara lebih detail dengan mengacu pada garis yang telah tergambar pada grafik. Dapat dilihat pada gambar 4.15 yang menggambarkan grafik jarak dan gambar 4.16 yang menggambarkan grafik tarif.



Gambar 4.15 Grafik perbandingan jarak

Pada gambar 4.15 digambarkan grafik dengan parameter jarak dan *test case*. Dengan penggunaan warna biru pada pelana sebagai penanda hasil jarak yang ada pada aplikasi yang telah dibuat, serta warna hijau penanda hasil jarak dari gojek. Dari proses *testing* yang telah dilakukan pada jarak didapatkan bahwa jarak yang ditempuh dari gojek memiliki rata-rata jarak yang lebih jauh dari pada aplikasi yang telah dibangun dengan rata-rata jarak yang lebih pendek. Pada *testing* jarak terdekat dapat dilihat bahwa gojek dapat memperkirakan jarak lebih baik dengan selisih 0.3 km, namun ketika *testing* pada jarak yang cukup jauh aplikasi yang dibuat dapat lebih baik dalam memperkirakan jarak dengan selisih 3 sampai 4 km dari jarak yang ada pada gojek.



Gambar 4.16 Grafik perbandingan tarif harga

Kemudian pada gambar 4.16 digambarkan grafik dengan parameter tarif dan *test case*. Dengan penggunaan warna biru pada pelana sebagai penanda hasil tarif yang ada pada aplikasi yang telah dibuat, serta warna hijau penanda hasil tarif dari gojek. Dari proses *testing* yang telah dilakukan pada tarif didapatkan bahwa tarif yang ditempuh dari gojek memiliki rata-rata tarif yang lebih mahal dari pada aplikasi yang telah dibangun dengan rata-rata tarif yang lebih murah. Pada *testing* tarif dapat dilihat bahwa pada aplikasi yang dibuat dapat lebih baik dalam memperkirakan tarif dengan selisih rata-rata Rp.245 /km dari tarif yang ada pada gojek.

4.2.3 Hasil Analisa

Dari hasil penelitian ini, didapatkan bahwa pengimplementasian metode *hevarsine* dapat memperkirakan jarak dan medan yang akan dilewati oleh kendaraan dalam perjalanan menuju tujuan pengguna. Pengimplementasi *lbs* pada proses *tag maps* dan *haversine* pada proses perkiraan jarak dan medan pada *maps* dirasa memiliki keakuratan yang

cukup baik untuk digunakan pada keadaan sebenarnya. Dengan didapatkannya jarak rute yang lebih pendek dari perkiraan yang dimunculkan oleh *google maps*, dengan hasil akurasi yang didapatkan dari proses *testing* kurang lebih 98% dalam memperkirakan jarak rute yang akan ditempuh diharapkan dapat membantu untuk meningkatkan efektifitas dari estimasi waktu yang dibutuhkan dalam menempuh rute. Dan dapat diperkirakan medan dengan akurasi hasil akurasi yang didapatkan dari hasil *testing* kurang lebih 95% diharap dapat memberikan gambaran kondisi keadaan medan yang akan dilewati. Dan dari proses *testing* perbandingan data jarak dan harga dengan membandingkan antara aplikasi yang telah dibangun dengan aplikasi gojek yang telah ada, untuk mengetahui akurasi dan efektifitas dari implementasi. Adapun data hasil rata-rata yang didapatkan dari perbandingan jarak adalah 0.32 km dan data hasil rata-rata yang didapatkan dari perbandingan harga adalah Rp.242, kemudian persentase akurasi yang didapat dari perbandingan yang dilakukan dari data varian jarak adalah 0.3 % dan persentase pada harga 0.24 %. Dari pengujian yang dilakukan menandakan bahwa aplikasi yang dibangun memiliki akurasi dan efektifitas yang sedikit lebih baik.

4.3 Integrasi Islam

Sistem yang dibangun dapat membantu memudahkan proses aktifitas manusia untuk berpindah dari satu tempat ketempat lain. Terkadang banyak oknum yang menyediakan jasa pelayanan transportasi tanpa standarisasi yang

jasas sehingga tidak terwujudnya keadilan. Dan jasa yang disediakan belum memiliki akurasi terkait rekomendasi baik rute maupun medan sekaligus pemilihan jenis kendaraan yang sesuai. Dengan adanya sistem ini, diharapkan dapat membantu para pengguna jasa transportasi yang akan melakukan aktifitas dengan menggunakan jasa transportasi yang adil dalam takaran transaksinya serta efektif dalam penggunaan estimasi waktu.

Hal ini sesuai dengan firman Allah SWT, dalam Al-Qur'an surat An-Nahl Ayat 90:

إِنَّ اللَّهَ يَأْمُرُ بِالْعَدْلِ وَالْإِحْسَانِ وَإِيتَايَ ذِي الْقُرْبَىٰ وَيَنْهَىٰ عَنِ الْفَحْشَاءِ وَالْمُنْكَرِ وَالْبَغْيِ
يُعْظِمُ لَكُمْ لَعْنَكُمْ تَذَكَّرُونَ ﴿٩٠﴾

“Sesungguhnya Allah menyuruh (kamu) berlaku adil dan berbuat kebajikan, memberi kepada kaum kerabat, dan Allah melarang dari perbuatan keji, kemungkaran dan permusuhan. Dia memberi pengajaran kepadamu agar kamu dapat mengambil pelajaran.” (QS. An-Nahl:90).

Penjelasan tafsir dari surat An-Nahl ayat 90 yang menjelaskan tentang konsep keadilan dari Allah untuk diimplementasikan oleh manusia agar kehidupan tetap berjalan adil, bila dikutip dari tafsir Jalalayn yang tertulis seperti berikut “(Sesungguhnya Allah menyuruh kalian berlaku adil) bertauhid atau berlaku adil dengan sesungguhnya (dan berbuat kebaikan) menunaikan fardu-fardu, atau hendaknya kamu menyembah Allah seolah-olah kamu melihat-Nya sebagaimana yang telah dijelaskan oleh hadis (memberi) bantuan (kepada kaum kerabat) famili mereka disebutkan secara khusus di sini, sebagai pertanda bahwa mereka harus dipentingkan terlebih dahulu (dan Allah melarang dari perbuatan keji) yakni zina (dan kemungkaran) menurut hukum

syariat, yaitu berupa perbuatan kekafiran dan kemaksiatan (dan permusuhan) menganiaya orang lain. Lafal al-baghyu disebutkan di sini secara khusus sebagai pertanda, bahwa ia harus lebih di jauhi dan demikian pula halnya dengan penyebutan lafal al-fahsyaa (Dia memberi pengajaran kepada kalian) melalui perintah dan larangan-Nya (agar kalian dapat mengambil pelajaran) mengambil pelajaran dari hal tersebut. Di dalam lafal tadzakkaruuna menurut bentuk asalnya ialah huruf ta-nya diidghamkan kepada huruf dzal. Di dalam kitab Al-Mustadrak disebutkan suatu riwayat yang bersumber dari Ibnu Masud yang telah mengatakan, bahwa ayat ini yakni ayat 90 surah An-Nahl, adalah ayat yang paling padat mengandung anjuran melakukan kebaikan dan menjauhi keburukan di dalam Al-Qur'an.”.

Kemudian dapat dilihat tentang konsep berniaga yang ada dalam Al-Qur'an. Adapun beberapa penggalan ayat yang menjelaskan konsep berniaga yang sesuai dengan konsep yang ada dalam islam. Adapun hal tersebut seperti yang dijelaskan pada Al-Qur'an surat Al-Baqarah ayat 198.

لَيْسَ عَلَيْكُمْ جُنَاحٌ أَنْ تَبْتَغُوا فَضْلًا مِنْ رَبِّكُمْ فَإِذَا أَفَضْتُمْ مِنْ عَرَفَاتٍ
فَاذْكُرُوا اللَّهَ عِنْدَ الْمَشْعَرِ الْحَرَامِ وَاذْكُرُوهُ كَمَا هَدَيْتُمْ
وَإِنْ كُنْتُمْ مِنْ قَبْلِهِ لَمَنِ الضَّالِّينَ ﴿١٩٨﴾

“Tidak ada dosa bagimu untuk mencari karunia (rezeki hasil perniagaan) dari Tuhanmu. Maka apabila kamu telah bertolak dari 'Arafat, berdzikirlah kepada Allah di Masy'arilharam. Dan berdzikirlah (dengan menyebut) Allah sebagaimana yang ditunjukkan-Nya kepadamu dan sesungguhnya kamu

sebelum itu benar-benar termasuk orang-orang yang sesat.” (Q.S Al-Baqarah: 198).

Penjelasan tafsir dari surat Al-Baqarah ayat 198 yang menjelaskan tentang konsep bagaimana cara berniaga yang sesuai dengan konsep yang ada dalam islam, bila dikutip dari tafsir Jalalayn yang tertulis seperti berikut “(Tidak ada dosa bagi kamu) dalam (mencari) atau mengusahakan (karunia) atau rezeki (dari Tuhanmu) yakni dengan berniaga di musim haji. Ayat ini turun untuk menolak anggapan mereka yang keliru itu (Maka jika kamu telah bertolak), artinya berangkat (dari Arafah) yakni setelah wukuf di sana, (maka berzikirlah kepada Allah), yakni setelah bermalam di Muzdalifah sambil membaca talbiah, tahlil dan berdoa (di Masyarilharam) yaitu nama sebuah bukit di ujung Muzdalifah disebut Quzah. Dalam sebuah hadis disebutkan bahwa Nabi saw. wukuf di sana, berzikir dan berdoa kepada Allah hingga hari telah amat benderang." (H.R. Muslim). (Dan berzikirlah kepada-Nya disebabkan petunjuk yang diberikan-Nya kepadamu) untuk mengetahui pokok-pokok agama dan tata cara hajinya. 'Kaf' menunjukkan sebab atau motifnya. (Dan sesungguhnya) dibaca 'in' bukan 'inna' (kamu sebelum itu) maksudnya sebelum petunjuk itu (termasuk orang-orang yang sesat).”.

Selanjutnya pada penggalan ayat yang menjelaskan hal serupa tentang penciptaan waktu dan keadaan terbaik dalam menjalani waktu dalam Al-Qur'an yaitu surat Al-'Asr ayat 1 sampai 3 yang berbunyi seperti berikut :

وَالْعَصْرِ ۝١ إِنَّ الْإِنْسَانَ لِرَبِّهِ خَسِيرٌ ۝٢ إِلَّا الَّذِينَ ءَامَنُوا وَعَمِلُوا الصَّالِحَاتِ ۝٣ وَتَوَاصَوْا بِالْحَقِّ وَتَوَاصَوْا بِالصَّبْرِ ۝٤

“Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran.”.(Q.S Al-'Asr :1-3).

Penjelasan tafsir dari surat Al-'Asr ayat 1 sampai 3 yang menjelaskan tentang konsep diciptakannya waktu dan keadaan terbaik dalam menjalaninya, bila dikutip dari tafsir Jalalayn yang tertulis seperti berikut “(Demi masa) atau zaman atau waktu yang dimulai dari tergelincirnya matahari hingga terbenamnya, maksudnya adalah waktu shalat asar. (Sesungguhnya manusia itu) yang dimaksud adalah jenis manusia (benar-benar berada dalam kerugian) di dalam perniagaannya. (Kecuali orang-orang yang beriman dan mengerjakan amal saleh) mereka tidak termasuk orang-orang yang merugi di dalam perniagaannya (dan nasihat-menasihati) artinya sebagian di antara mereka menasihati sebagian yang lainnya (supaya menaati kebenaran) yaitu iman (dan nasihat-menasihati dengan kesabaran) yaitu di dalam menjalankan amal ketaatan dan menjauhi kemaksiatan.”

Maka dari itu beberapa aspek tersebut sehubungan dengan pembahasan implementasi perhitungan yang memperkirakan jarak dan medan, dimana penentuan tersebut haruslah adil dan sesuai dengan prosedur yang ada, yakni benar-benar menerapkan kaidah yang bersesuaian pada saat melakukan perhitungan jarak atau medan. Keadaan penciptaan bumi, bentuk kendaraan yang digunakan menjadi penunjang, serta kebutuhan manusia akan kendaraan untuk menunjang aktifitas yang dilakukan dirasa telah dicakup oleh beberapa

penggalan ayat-ayat dari beberapa surat yang telah dikutip dari Al-Qur'an dan ditafsirkan berdasarkan kitab tafsir yang telah distandarkan dalam ilmu tafsir Al-Qur'an.



BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian dan pembahasan tentang aplikasi *mobile online* menggunakan *lbs* dan metode *haversine* dalam pengimplementasian jarak dan medan, dapat diambil kesimpulan. Pencarian kecocokan pada saat *testing* perhitungan *manual* yang dilakukan dengan beberapa tahap yakni pengambilan *latitude* dan *logintude* kemudian diturunkan korelasinya kedalam bentuk busur sehingga dapat dihitung kontur kemiringannya, Berdasarkan hasil uji coba perhitungan pada data lokasi yang telah dilakukan. Kemudian dari tahap *testing* tersebut dapat diimplementasikan kedalam beberapa fungsi antara lain proses *auto complete lacion* berjalan sesuai dengan keadaan sebenarnya, proses penggambaran rute berdasarkan jarak yang telah dihitung dengan akurasi persentasi dari metode 98% untuk jarak dan 95% untuk medan, sementara hasil persentase perbandingan akurasi aplikasi untuk akurasi jarak 0.3% serta 0.24% untuk perbandingan estimasi harga.

Rekomendasi penumpang atau *driver* terdekat, berdasarkan daerah asal dan tujuan mempunyai letak dan koordinat yang berdekatan, sehingga jarak jemput tidak terlalu jauh. Selanjutnya *system* akan memberikan nampilkan biaya, jarak, rute serta jalan-jalan yang akan dilalui. medan dapat dicek ketika sesuai dengan lokasi yang dituju. *Pos-pos driver* yang dipetakan pada *maps* dapat menjadi rekomendasi alternatif untuk memilih *driver*.

5.2 Saran

Berdasarkan pengujian yang telah dilakukan pada *Mobile App online* menggunakan *lbs* dan metode *haversine* dalam pengimplementasian jarak dan medan, tentunya masih banyak kekurangan dan kelemahan yang ada pada aplikasi sehingga perlu pengembangan lebih lanjut. Untuk pengembangan sistem ini, diperlukan penambahan integrasi dengan pihak ke 3 untuk memudahkan proses transaksi layanan dan jasa. Aspek pertimbangan dalam mengembangkan aplikasi pada aspek penggunaan akses memory pada perangkat lunak yang menjalankan aplikasi dapat diperingan beban dan kinerjanya.

DAFTAR PUSTAKA

- Al-Qurtubi, Muhammad (620 H) "Al-Jami' liahkam al-Qur'an wa al-Mubayyin Lima Tadhammanahu Min as-Sunnah wa Ayi al-Furqan"
- Braun, S., Elberzhager, F., Holl, K., 2017. Automation Support for Mobile App Quality Assurance – A Tool Landscape. *Procedia Computer Science* 110, 117–124. <https://doi.org/10.1016/j.procs.2017.06.129>
- Chao Lin Liu, T. W. (2002). Path Planning Algorithms for Public Transportation System. The Fourth International IEEE Conference on Intelligent Transportation Systems. Oakland, California, USA.
- Chopde, N.R., Nichat, M.K., 2013. Landmark based shortest path detection by using A* and Haversine formula. *International Journal of Innovative Research in Computer and Communication Engineering* 1, 298–302.
- Fechner, P., Boudier, T., Mangenot, S., Jaroslowski, S., Sturgis, J.N., Scheuring, S., 2009. Structural Information, Resolution, and Noise in High-Resolution Atomic Force Microscopy Topographs. *Biophysical Journal* 96, 3822–3831. <https://doi.org/10.1016/j.bpj.2009.02.011>
- Geisberger, R. (2011). Advanced Route Planning in Public Transportation. Munchen: Karlsruher Instituts für Technologie.
- Gintoro, dkk. 2010, Analisis dan Perancangan Sistem Pencarian Taksi Terdekat dengan Pelanggan Menggunakan Layanan Berbasis Lokasi, Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010), Yogyakarta.
- Hsiao, Ying-Tung, Cheng-Long Chuang dan Cheng-Chih Chien (2004). Ant Colony Optimization for Best Path Planning. Taipei: Tamkang University.
- Mahapoonyanont, N., Mahapoonyanont, T., Pengkaew, N., Kamhangkit, R., 2010. Power of the test of One-Way Anova after transforming with large sample size data. *Procedia - Social and Behavioral Sciences* 9, 933–937. <https://doi.org/10.1016/j.sbspro.2010.12.262>
- Moh, Ali Mustofa "Pencarian Lokasi Hotel Berbasis Android Menggunakan metode Location Based Service (Studi Kasus Kota Semarang)"
- Morgado, I.C., Paiva, A.C.R., 2016. Impact of Execution Modes on Finding Android Failures. *Procedia Computer Science* 83, 284–291. <https://doi.org/10.1016/j.procs.2016.04.127>
- Muntenescu, Florina 2017. Espresso Testing Android Developer, <<https://developer.android.com/training/testing/espresso>>
- Odijk D (2002) Fast Precise GPS Positioning in the Presence of Ionospheric Delays,

PhD Thesis, Delft University of Technology, Delft, The Netherlands, 69-102.

Pressman, R.S. (2010), Software Engineering : a practitioner's approach, McGraw-Hill, New York, 68.

Safaat H, Nazruddin 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika Bandung: Bandung

