

**PENGEMBANGAN SISTEM REKOMENDASI RUTE GOWES DI
KOTA MALANG BERBASIS ANDROID**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ahmad Aulia Fahmi

NIM: 155150201111387



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

PENGEMBANGAN SISTEM REKOMENDASI RUTE GOWES DI KOTA MALANG
BERBASIS ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ahmad Aulia Fahmi
NIM: 155150201111387

Skripsi ini telah diuji dan dinyatakan lulus pada
8 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ratih Kartika Dewi, S.T., M.Kom
NIK: 201503 890520 2 001

Lutfi Fanani, S.Kom., M.T., M.Sc.
NIK: 2016078902171001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan , S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 20 Juni 2019

Ahmad Aulia Fahmi

NIM: 155150201111387



PRAKATA

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan Rahmat, Taufik dan Hidayah – Nya sehingga skripsi yang berjudul “Pengembangan Sistem Rekomendasi Rute Gowes di Kota Malang Berbasis Android” dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Ratih Kartika Dewi, S.T.,M.Kom selaku dosen pembimbing 1 yang telah mengarahkan, membimbing serta memberikan semangat dalam penyelesaian skripsi ini.
2. Bapak Lutfi Fanani, S.Kom., M.T., M.Sc. selaku dosen pembimbing 2 yang telah memberikan masukan, ilmu serta saran dalam penyelesaian skripsi ini.
3. Keluarga yang telah memberikan dukungan serta semangat sehingga penulis dapat menyelesaikan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua jurusan Teknik Informatika.
5. Teman – teman jurusan Teknik Informatika angkatan 2015 yang selalu memberikan dukungan dalam penyelesaian skripsi ini.
6. Pihak – pihak lain yang terlibat secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 20 Juni 2019

Penulis

ahmadauliahfahmi@student.ub.ac.id

ABSTRAK

Ahmad Aulia Fahmi, Pengembangan Sistem Rekomendasi Rute Gowes di Kota Malang Berbasis Android

Pembimbing: Ratih Kartika Dewi, S.T.,M.Kom dan Lutfi Fanani, S.Kom., M.T., M.Sc.

Bersepeda merupakan salah satu aktivitas olahraga yang menjadi pilihan sebagian masyarakat Indonesia di kota – kota besar. Bersepeda sendiri dilakukan dengan mengayuh pedal sepeda dengan tenaga serta ritme yang teratur. Salah satu kota dengan peminat olahraga sepeda di Indonesia adalah Kota Malang Jawa Timur. Namun, bagi para pemula yang memiliki minat pada bidang bersepeda memiliki kesulitan dalam menemukan rute bersepeda di Kota Malang. Dengan menggunakan aplikasi rekomendasi rute gowes ini pengguna dapat menemukan rute bersepeda sesuai dengan keinginan mereka.

Sistem rekomendasi rute gowes di Kota Malang ini dirancang dengan menggunakan metode TOPSIS (Technique For Order Preference by Similarity to Ideal Solution) menggunakan kriteria jarak rute sepeda, waktu tempuh rute, serta kesulitan medan rute sepeda dan di implementasikan pada sistem operasi Android. Pengujian untuk sistem ini menggunakan tiga cara yaitu pengujian blackbox, pengujian validasi algoritma dan pengujian usability. Pengujian blackbox adalah pengujian dengan melihat apakah hasil keluaran sistem sama dengan hasil yang diharapkan. Pada pengujian blackbox ini menghasilkan nilai 100% atau semua fungsionalitas pada sistem sudah sesuai dengan yang diharapkan. Pengujian validasi algoritme ialah pengujian yang bertujuan untuk membandingkan hasil keluaran sistem dengan kalkulasi manual. Pengujian validasi algoritme pada penelitian ini menghasilkan nilai 100% atau hasil keluaran sistem sama dengan kalkulasi manual. Sedangkan pengujian usability adalah pengujian antarmuka sistem yang dilakukan oleh pengguna akhir. Hasil pengujian usability ini menghasilkan nilai 84,5 dari skala yang telah digunakan (memiliki grade B) atau dapat dikatakan sistem ini bernilai baik oleh pengguna akhir.

Kata kunci: sistem rekomendasi,gowes,TOPSIS,pengujian usability,pengujian blackbox

ABSTRACT

Ahmad Aulia Fahmi, Pengembangan Sistem Rekomendasi Rute Gowes di Kota Malang Berbasis Android

Supervisors: Ratih Kartika Dewi, S.T.,M.Kom and Lutfi Fanani, S.Kom., M.T., M.Sc.

Cycling has become one of the few favorite sports activities for many city dwellers in Indonesia. Cycling is the activity of pedaling on a bicycle with a certain rhythm. One of the cities with a significant amount of cyclists is Malang City, located in the province of East Java. However, for certain individuals, such as youngsters, have had difficulty finding cycling routes in Malang city. An application that recommends cycling routes is capable of providing cycling routes based on a user's preference.

The cycling route recommendation application in Malang city is designed using the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) method which used criteria such as the distance of a cycling route, traveling time of the route, and difficulty of the terrain and implemented on an Android operating system. The application underwent three testing methods which were blackbox testing, algorithm validation testing and usability testing. Blackbox testing is the process of observing if the system output is the same as the intended output. The results from the blackbox method resulted in a 100% score, which is the equivalent of all functionalities fulfilling expectations. Testing the algorithm validation from testing is intended to compare the results of the system output with manual calculations. Testing the algorithm validation in this study produces a value of 100% or the system output is the same as manual calculation. Whereas the usability is tested thru the system's interface which was tested by users. The usability testing method resulted in a score of 84,5 from a predetermined scale (have grade B) and is equivalent to stating that the system was given a high score by the users.

Keywords: recommendation system, paddling, TOPSIS, usability testing, blackbox testing

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Android	7
2.2.1 Linux Kernel.....	8
2.2.2 Hardware Abstraction Layer (HAL)	9
2.2.3 Android Runtime	9
2.2.4 Native C/C++ Libraries.....	9
2.2.5 Java API Framework	9
2.2.6 System Apps	9
2.3 GPS <i>Smartphone</i>	10
2.4 Google Maps	10
2.5 Google Firebase	11
2.6 Metode TOPSIS	12
2.6.1 Pengertian TOPSIS.....	12

2.6.2 Langkah – langkah Perhitungan TOPSIS.....	12
2.7 Pengujian Perangkat Lunak.....	13
2.7.1 Pengujian Blackbox	14
2.7.2 Pengujian <i>Usability</i>	14
BAB 3 METODOLOGI	15
3.1 Studi Literatur	16
3.2 Analisis Kebutuhan Sistem.....	16
3.3 Perancangan	16
3.3.1 Perancangan Sistem	16
3.3.2 Perancangan Algoritme.....	17
3.4 Implementasi Sistem	17
3.5 Analisis dan Pengujian Sistem	17
3.6 Kesimpulan dan Saran	17
BAB 4 Analisis dan Perancangan	18
Gambaran Aplikasi	18
4.1 Analisis Kebutuhan	19
4.1.1 Identifikasi Aktor	20
4.1.2 Analisis Kebutuhan Fungsional	20
4.1.3 Analisis Kebutuhan Non-Fungsional	21
4.1.4 <i>Use Case Diagram</i>	22
4.1.5 <i>Use Case Scenario</i>	22
4.2 Perancangan	25
4.2.1 Perancangan <i>Activity Diagram</i>	25
4.2.2 Perancangan <i>Sequence Diagram</i>	28
4.2.3 Perancangan <i>Class Diagram</i>	31
4.2.4 Perancangan Basis Data	31
4.2.5 Perancangan Antarmuka.....	32
4.2.6 Perancangan Algoritme.....	34
BAB 5 IMPLEMENTASI	40
5.1 Spesifikasi Sistem	40
5.1.1 Spesifikasi Perangkat Keras.....	41
5.1.2 Spesifikasi Perangkat Lunak	41

5.2 Batasan – batasan Implementasi.....	42
5.3 Implementasi Basis Data.....	42
5.4 Implementasi <i>Class</i> dan <i>Assets</i>	45
5.5 Implementasi Kode Program	45
5.5.1 Implementasi Kode Program Memberikan Bobot Nilai.....	46
5.5.2 Implementasi Kode Program Melihat Rekomendasi Rute Gowes.....	49
5.5.3 Implementasi Kode Program Melihat Detail Rekomendasi	51
5.5.4 Implementasi Kode Program Melihat Petunjuk Arah.....	53
5.5.5 Implementasi Kode Program Perhitungan TOPSIS	55
5.6 Implementasi Antarmuka Sistem	62
BAB 6 PENGUJIAN	65
6.1 Pengujian <i>Blackbox</i>	65
6.2 Pengujian Validasi Algoritme	69
6.3 Pengujian <i>Usability</i>	71
6.4 Analisis dan Hasil Pengujian.....	73
BAB 7 KESIMPULAN DAN SARAN	75
7.1 Kesimpulan.....	75
7.2 Saran	75
DAFTAR REFERENSI	76
LAMPIRAN A DATA SURVEI PEMINAT <i>GOWES</i> DI KOTA MALANG.....	80
LAMPIRAN B <i>CLASS DIAGRAM</i>	82
LAMPIRAN C DATA RUTE <i>GOWES</i> DI KOTA MALANG	83

DAFTAR TABEL

Tabel 2.1 Penjelasan Penelitian Terdahulu	7
Tabel 4.1 Identifikasi Aktor	20
Tabel 4.2 Kebutuhan Fungsional Sistem	20
Tabel 4.3 Kebutuhan Non-Fungsional Sistem	21
Tabel 4.4 Memberikan Bobot Nilai Untuk Setiap Kriteria.....	23
Tabel 4.5 Melihat Rekomendasi Rute Gowes	23
Tabel 4.6 Melihat Detail Rekomendasi Rute.....	24
Tabel 4.7 Melihat Petunjuk Arah	24
Tabel 4.8 Perancangan Basis Data	31
Tabel 4.9 Penjelasan Kriteria Penelitian	36
Tabel 4.10 Alternatif Data	36
Tabel 4.11 Normalisasi Alternatif.....	37
Tabel 4.12 Bobot Nilai Kriteria	37
Tabel 4.13 Normalisasi Terbobot	38
Tabel 4.14 Solusi Ideal Positif dan Negatif.....	38
Tabel 4.15 Jarak Solusi Ideal Positif dan Negatif.....	39
Tabel 4.16 Nilai Preferensi	39
Tabel 5.1 Spesifikasi Perangkat Keras Komputer	41
Tabel 5.2 Spesifikasi Perangkat Keras <i>Android</i>	41
Tabel 5.3 Spesifikasi Perangkat Lunak Komputer	41
Tabel 5.4 Spesifikasi Perangkat Lunak <i>Android</i>	42
Tabel 5.5 Implementasi Basis Data	42
Tabel 5.6 Implementasi <i>Class</i> dan <i>Assets</i>	45
Tabel 5.7 Implementasi Kode Program Memberikan Bobot Nilai	46
Tabel 5.8 <i>Method setOnClickListener</i> pada <i>Class</i> Daftar_Rute.....	48
Tabel 5.9 Implementasi Kode Program Melihat Rekomendasi Rute	49
Tabel 5.10 <i>Method onBindViewHolder</i>	50
Tabel 5.11 Implementasi Kode Program Melihat Detail Rute	52
Tabel 5.12 <i>Method onClickListener</i> pada <i>Class</i> Detail_Rute.....	54
Tabel 5.13 Inisialisasi <i>Firestore</i> dan <i>ArrayList</i>	55

Tabel 5.14 <i>Method getDataRute</i>	56
Tabel 5.15 <i>Method TOPSIS</i>	58
Tabel 6.1 Pengujian Fungsionalitas Memberikan Bobot Nilai Untuk Setiap Kriteria	65
Tabel 6.2 Pengujian Fungsionalitas Melihat Rekomendasi Rute Gowes	66
Tabel 6.3 Pengujian Fungsionalitas Melihat Detail Rekomendasi Rute.....	67
Tabel 6.4 Pengujian Fungsionalitas Melihat Petunjuk Arah	68
Tabel 6.5 Bobot Nilai Pengujian Validasi Algoritme.....	70
Tabel 6.6 Data Rute Gowes	70
Tabel 6.7 Perbandingan Perhitungan Sistem dengan Perhitungan Manual.....	71
Tabel 6.8 Sebaran Jawaban Responden Terhadap Kuisisioner	72
Tabel 6.9 Hasil Penilaian Responden	73



DAFTAR GAMBAR

Gambar 1.1 Fitur Google Maps	2
Gambar 2.1 Arsitektur Sistem Operasi Android	8
Gambar 2.2 Rekomendasi Rute dengan Google Maps	11
Gambar 2.3 Logo Google Firebase	11
Gambar 2.4 Penentuan Hasil Penilaian <i>System Usability Scale (SUS)</i>	14
Gambar 3.1 Diagram Alir Metodologi	15
Gambar 4.1 Diagram Alir Analisis dan Perancangan.....	18
Gambar 4.2 Gambaran Aplikasi	19
Gambar 4.3 Penjelasan Kode Kebutuhan Sistem.....	20
Gambar 4.4 Use Case Diagram Sistem Rekomendasi rute Gowes	22
Gambar 4.5 <i>Activity Diagram</i> Memberikan Bobot Nilai Untuk Setiap Kriteria	26
Gambar 4.6 <i>Activity Diagram</i> Melihat Rekomendasi Rute Gowes	27
Gambar 4.7 <i>Activity Diagram</i> Melihat Detail Rekomendasi Rute	27
Gambar 4.8 <i>Activity Diagram</i> Melihat Petunjuk Arah	28
Gambar 4.9 <i>Sequence Diagram</i> Memberikan Bobot Nilai Untuk Setiap Kriteria .	29
Gambar 4.10 <i>Sequence Diagram</i> Melihat Rekomendasi Rute Gowes.....	30
Gambar 4.11 <i>Sequence Diagram</i> Melihat Detail Rekomendasi Rute	30
Gambar 4.12 <i>Class Diagram</i>	31
Gambar 4.13 Antarmuka Memberikan Bobot Nilai Untuk Setiap Kriteria	33
Gambar 4.14 Antarmuka Melihat Rekomendasi Rute Gowes	33
Gambar 4.15 Antarmuka Melihat Detail Rekomendasi Rute dan Petunjuk Arah	34
Gambar 4.16 <i>Flowchart</i> TOPSIS	35
Gambar 5.1 Diagram Alir Implementasi.....	40
Gambar 5.2 Implementasi Antarmuka Beranda	62
Gambar 5.3 Implementasi Antarmuka Fungsionalitas Memberikan Bobot Nilai .	63
Gambar 5.4 Implementasi Antarmuka Fungsionalitas Melihat Rekomendasi Rute Gowes.....	63
Gambar 5.5 Implementasi Antarmuka Fungsionalitas Melihat Detail Rute	64
Gambar 5.6 Implementasi Antarmuka Fungsionalitas Melihat Petunjuk Arah	64
Gambar 6.1 Diagram Alir Pengujian.....	65

Gambar 6.6 Penentuan Hasil Penilaian.....	74
---	----



DAFTAR LAMPIRAN

LAMPIRAN A DATA SURVEI PEMINAT <i>GOWES</i> DI KOTA MALANG	80
LAMPIRAN B <i>CLASS DIAGRAM</i>	82
LAMPIRAN C DATA RUTE <i>GOWES</i> DI KOTA MALANG	83



BAB 1 PENDAHULUAN

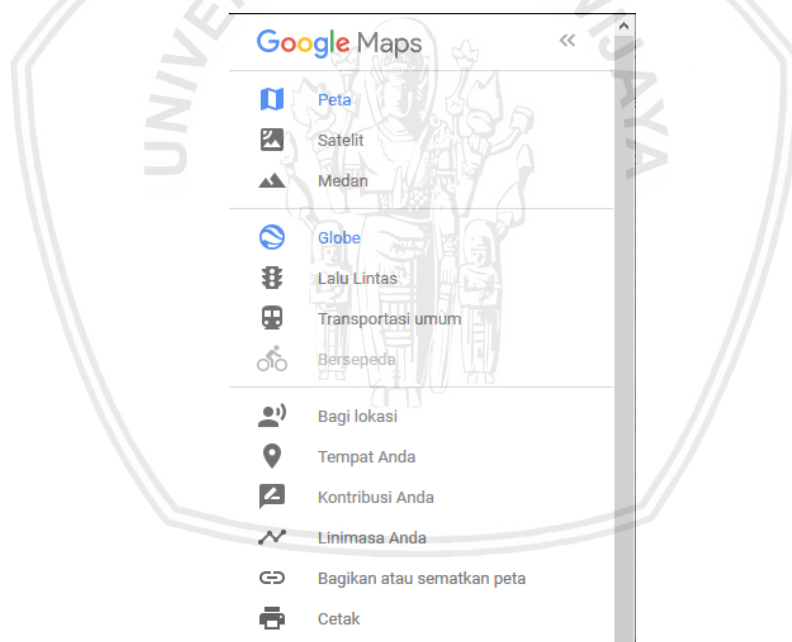
1.1 Latar Belakang

Bersepeda yakni pilihan aktivitas olahraga bagi sebagian masyarakat Indonesia di kota – kota besar. Bersepeda sendiri dilakukan dengan mengayuh pedal sepeda dengan tenaga serta ritme yang teratur. Pada saat ini bersepeda memiliki istilah lain dalam masyarakat modern yaitu *gowes* yang memiliki arti mengayuh atau menggowes (KBBI,2016). Bersepeda sendiri memiliki beberapa manfaat diantaranya adalah untuk mencegah resiko kanker,mengurangi tingkat stress seseorang,melancarkan aliran darah,menyehatkan jantung dan sebagainya (Depkes,2017). Pada saat ini,olahraga sepeda pun bisa dilakukan untuk menggantikan mode transportasi,contohnya saja bersepeda pada saat berangkat sekolah atau kantor. Dengan menggunakan waktu tempuh yang ada,seseorang juga bisa menikmati olahraga tanpa harus mengeluarkan waktu yang banyak.

Bersepeda sendiri merupakan jenis olahraga aerobik. Olahraga aerobik ini sendiri adalah kegiatan olahraga yang mengeluarkan sedikit energi sampai banyak energi secara berkesinambungan. Aktivitas aerobik yakni salah satu kegiatan olahraga yang bersandar kepada ketersediaan oksigen untuk melakukan proses metabolisme dalam tubuh. Sehingga dalam prosesnya,aktivitas olahraga aerobik ini bergantung kepada kinerja yang ideal dari bagian – bagian dalam tubuh seperti paru – paru, jantung serta pembuluh darah. Manfaat yang bisa didapatkan dari bersepeda ini antara lain dapat memperkuat dan menyehatkan jantung. Hal ini dikarenakan pada saat aktivitas bersepeda berlangsung,peran jantung adalah memompa kebutuhan oksigen dalam jumlah yang besar ke seluruh tubuh. Manfaat kedua yaitu dapat melancarkan peredaran darah serta dapat menyusutkan bobot badan. Proses tersebut disebabkan oleh bertambahnya elastisitas pembuluh darah yang dikarenakan berkurangnya tumpukan lemak akibat dari simpanan lemak lebih banyak dibakar. Dampak positif dari proses tersebut menjadikan nilai LDL atau *Low Density Lipoprotein* menurun, nilai HDL atau *High Density Lipoprotein* meningkat, mengakibatkan bobot badan relatif proporsional. Manfaat lainnya adalah meningkatkan elastisitas paru – paru dimana kemampuan paru – paru untuk berkontraksi menjadi bertambah (Palar,2015).

Sistem rekomendasi ialah salah satu jenis aplikasi yang menampilkan keputusan pengamatan atas situasi serta kepentingan pengguna(Wahyo,2015). Oleh karenanya,sistem rekomendasi membutuhkan jenis anjuran yang sesuai supaya keputusan yang dianjurkan sama dengan kepentingan pengguna, selain itu untuk meringankan pengguna mengambil pertimbangan yang sesuai dalam memutuskan barang yang akan digunakan(Wahyo,2015). Terdapat beberapa contoh sistem rekomendasi diantaranya yakni sistem rekomendasi pemilihan tempat wisata di Kota Banyuwangi. Sistem ini dapat menyediakan anjuran lokasi wisata yang berada di Kota Banyuwangi dengan masukan pengguna berupa kategori wisata,kategori biaya, moda transportasi, kegiatan wisata serta letak

pengguna (Kusuma,2017). Selain itu, sistem rekomendasi lain bisa diterapkan dalam sebuah pengambilan keputusan oleh suatu perusahaan untuk menentukan karyawan kontrak menjadi karyawan tetap. Sistem rekomendasi ini diteliti pada PT Gowa Motor Group dengan cara, pengguna memasukkan beberapa kriteria yang mendukung seorang karyawan ditetapkan menjadi karyawan tetap setelah menjadi karyawan kontrak. Ada pun kriteria tersebut adalah kinerja, kedisiplinan, loyalitas, pengalaman bekerja, dan tingkat Pendidikan (Mallu,2015). Salah satu sistem rekomendasi yang populer adalah Google Maps. Google Maps merupakan fasilitas denah digital milik Google yang bisa dimanfaatkan oleh pengembang aplikasi Android untuk disematkan kedalam aplikasi mereka. Google Maps secara otomatis akan mengirimkan data berupa tampilan peta serta respon pengguna berdasarkan permintaan pengguna langsung dari server Google (Google,2019). Pada Google Maps terdapat beberapa fitur diantaranya adalah pengguna bisa melihat peta dalam bentuk 2D dan 3D. Google Maps juga dapat memberikan rekomendasi rute perjalanan secara langsung. Google Maps mendukung rekomendasi rute menggunakan kendaraan pribadi serta dapat melihat situasi lalu lintas. Namun Google Maps saat ini belum mendukung adanya fitur untuk melihat rute perjalanan menggunakan sepeda seperti Gambar 1.1 :



Gambar 1.1 Fitur Google Maps

Sumber : google (2019)

Kota Malang merupakan kota yang memiliki luas daerah mencapai 110.06 km². Penduduk kota Malang sendiri terdiri dari berbagai etnis diantaranya adalah etnis Madura, Jawa dan beberapa keturunan Cina dan Arab (Pemerintah Kota Malang, 2019). Kota Malang juga merupakan salah satu kota dengan tingkat antusias terhadap sepeda yang cukup tinggi. Pada tahun 2015, kontingen balap sepeda Kota Malang pada ajang Pekan Olahraga Provinsi Jawa Timur V yang diselenggarakan di Banyuwangi menghasilkan 2 medali emas dan 3 medali perunggu yang masing – masing disumbangkan oleh Crismonita, Fajar dan M.

Praditya (Pemerintah Kota Malang,2015). Ditahun berikutnya, pemerintah kota malang menggelar kejuaraan BMX Piala Wali Kota Malang. Pada kejuaran tersebut tercatat 320 pendaftar mengikuti ajang tersebut (Pemerintah Kota Malang,2016). Sementara di tahun 2019, atlet sepeda kota malang siap tampil pada ajang Asian Track Clycling Championship (ACC) di Jakarta Timur(Pemerintah Kota Malang,2019). Dari hasil suvei yang telah peneliti lakukan (Lampiran A) terdapat 81,8 % responden kesulitan dalam mencari rute *gowes* di Kota Malang dengan beberapa kendala yang dialami berupa tidak mengetahui jalur sepeda, informasi yang minim, serta tidak mengetahui waktu tempuh dan medan yang dilalui.

Smartphone pada saat ini merupakan kebutuhan primer setiap orang. Hal ini bisa dilihat dari prediksi jumlah pengguna *smartphone* dari tahun 2016 sampai dengan 2019 menjumpai penambahan yang cukup substansial. Tahun 2016 pengguna *smartphone* diprediksi berjumlah 65,2 juta,di tahun berikutnya meningkat menjadi 74,9 juta,tahun 2018 pengguna *smartphone* sudah mencapai 83,5 juta dan pada tahun 2019 diperkirakan pengguna *smartphone* akan mencapai 92 juta orang di Indonesia (Katadata,2016). Android ialah sistem operasi berbasis *Linux* untuk perangkat bergerak seperti *smartphone* dan tablet(Maryam,2016). Di Indonesia terdapat beberapa sistem operasi untuk perangkat seluler diantaranya adalah sistem operasi Android,iOS,Blackberry OS,Nokia,SymbianOS dan lain sebagainya. Pada tahun 2018 – 2019 jumlah pengguna android sebesar 91.02% dari total seluruh pengguna *smartphone* di Indonesia. Selanjutnya pada urutan kedua disusul oleh pengguna iOS sebesar 4.41%, kemudian disusul oleh pengguna sistem operasi lainnya sebesar 2.55% (Statcounter,2019).

Software Development Life Cycle merupakan kerangka kerja atau dasar untuk menata, merancang, dan melaksanakan tugas – tugas yang berkaitan dengan pengembangan sistem (Lemke,2018). Tujuan dari *Software Development Life Cycle* ini sendiri adalah membantu pemangku kepentingan dalam pengembangan sistem untuk membuat sistem yang memenuhi persyaratan teknis dan memenuhi kebutuhan pengguna sistem (Lemke,2018). Dalam *Software Development Life Cycle*, terdapat beberapa fase diantaranya adalah inisiasi, konsep pengembangan sistem, perencanaan, analisis kebutuhan, perancangan, pengembangan, integrasi dan pengujian, implementasi, pemeliharaan serta pergantian (Lemke,2018). Dalam penelitian ini,penulis hanya berfokus pada fase merancang, mengimplementasikan serta menguji sistem rekomendasi rute *gowes* di kota Malang.

Berdasarkan penjelasan,data – data serta statistik seperti yang dijabarkan sebelumnya, maka penulis hendak mengembangkan sistem rekomendasi rute *gowes* di Kota Malang berbasis android. Sistem dibuat dengan menerapkan metode TOPSIS (*Technique for Order Performance of Similarity to Ideal Solution*). Penerapan metode TOPSIS pada penelitian ini karena kemudahan dalam penerapan serta rendahnya tingkat kompleksitas algoritma yang dipakai. Dan menggunakan sensor GPS yang terdapat pada *smartphone* saat ini untuk memudahkan pengguna mendapatkan lokasi rute *gowes* yang diinginkan. Dengan

adanya sistem ini, penulis berharap bisa membantu masyarakat Kota Malang dalam menentukan jalur atau rute terbaik untuk aktivitas bersepeda.

1.2 Rumusan Masalah

Berlandaskan permasalahan yang sudah dipaparkan oleh penulis pada latar belakang, maka rumusan masalah yang diharapkan dapat diselesaikan adalah sebagai berikut :

1. Bagaimana merancang sistem rekomendasi rute *gowes* di Kota Malang berbasis android?
2. Bagaimana cara mengimplementasikan sistem rekomendasi rute *gowes* di Kota Malang?
3. Bagaimana hasil pengujian dari sistem rekomendasi rute *gowes* di Kota Malang setelah diimplementasikan?

1.3 Tujuan

Dari rumusan masalah diatas, penulis memiliki tujuan sebagai berikut :

1. Merancang sistem rekomendasi rute *gowes* di Kota Malang berbasis android
2. Mengimplementasikan sistem rekomendasi rute *gowes* di Kota Malang
3. Menguji sistem rekomendasi rute *gowes* di Kota Malang setelah diimplementasikan

1.4 Manfaat

Dalam penelitian ini, penulis mengharapkan dapat membantu komunitas sepeda maupun masyarakat umum yang berada di kota Malang untuk memilih rute *gowes* secara mudah dengan kriteria yang ditentukan oleh pengguna sendiri.

1.5 Batasan Masalah

Berlandaskan rumusan masalah yang telah dijelaskan sebelumnya, penelitian ini memiliki beberapa batasan masalah agar dalam prosesnya penelitian ini bisa berfokus pada masalah apa saja yang sudah dijabarkan. Berikut adalah batasan – batasan masalah yang penulis jabarkan :

1. Penelitian ini diterapkan untuk merekomendasikan rute *gowes* hanya di kota Malang, Jawa Timur
2. Metode yang diterapkan adalah metode TOPSIS
3. Sensor yang dimanfaatkan adalah sensor GPS yang terdapat pada *smartphone* pengguna
4. Alternatif yang digunakan adalah rute *gowes* berdasarkan kriteria yang telah dipilih pengguna

5. Kriteria yang digunakan dalam menentukan rute gowes adalah jarak rute, waktu tempuh rute dan kesulitan medan pada rute gowes
6. Sistem rekomendasi rute gowes dikembangkan untuk sistem operasi minimal android versi 4.1 (Jelly Bean) dan maksimal sampai dengan versi 9 (Pie)

1.6 Sistematika Pembahasan

Sistematika penulisan dalam penelitian ini mencakup beberapa bab, diantaranya sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjabarkan mengenai latar belakang, rumusan masalah, tujuan, manfaat serta batasan masalah tentang “Pengembangan Sistem Rekomendasi Rute Gowes di Kota Malang Berbasis Android”.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan tentang dasar teori dan kajian pustaka sistem rekomendasi yang diimplementasikan pada penelitian ini. Kajian pustaka yang digunakan dalam penelitian ini berdasarkan penelitian terdahulu yang memiliki beberapa kesamaan. Teori yang digunakan adalah metode TOPSIS, *Google Firebase*, *Google Maps SDK*, GPS dan pengujian perangkat lunak.

BAB III METODELOGI PENELITIAN

Bab ini menjabarkan mengenai langkah kerja serta metode dalam pengembangan perangkat lunak “Pengembangan Sistem Rekomendasi Rute Gowes di Kota Malang Berbasis Android” yang mencakup metode wawancara, studi literatur, pengumpulan data, analisis kebutuhan pengguna, perancangan sistem, implementasi sistem, pengujian sistem dan kesimpulan serta saran.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini berisikan analisis kebutuhan dan perancangan sistem. Pada analisis kebutuhan mencakup *usecase* diagram dan *usecase* skenario, sedangkan dalam perancangan sistem meliputi perancangan antarmuka, *class diagram*, *sequence diagram*, *activity diagram*, perancangan basis data dan perancangan algoritma.

BAB V IMPLEMENTASI

Bab ini mencakup tahap – tahap implementasi sistem seperti batasan implementasi, spesifikasi sistem, implementasi basis data, implementasi antarmuka, implementasi kode, serta implementasi class serta *assets* dalam sistem.

BAB VI PENGUJIAN

Bab ini menjabarkan mengenai metode dan teknik pengujian yang diterapkan dalam penelitian ini. Teknik pengujian yang diterapkan dalam penelitian ini adalah teknik pengujian blackbox, perhitungan manual algoritma serta pengujian usability.

BAB VII PENUTUP

Bab ini memberikan kesimpulan serta saran setelah penelitian “Pengembangan Sistem Rekomendasi Rute Gowes di Kota Malang Berbasis Android” ini telah penulis selesaikan.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan ialah bab yang menguraikan kajian pustaka dan dasar teori yang diterapkan penulis untuk diimplementasikan ke dalam laporan penelitian. Kajian pustaka ialah pengkajian mengenai penelitian terdahulu oleh orang lain maupun kelompok ilmiah yang memiliki konsep penelitian yang sama dengan penulis. Sedangkan dasar teori ialah pengkajian perihal teori – teori dasar yang akan penulis gunakan dalam merancang sistem rekomendasi ini.

2.1 Kajian Pustaka

Penelitian terdahulu tentang pengembangan sistem rekomendasi serta penggunaan metode TOPSIS telah banyak dilakukan oleh peneliti terdahulu, penulis telah mendapatkan kajian pustaka yang relevan dengan konsep penelitian ini seperti terurai pada Tabel 2.1 berikut ini :

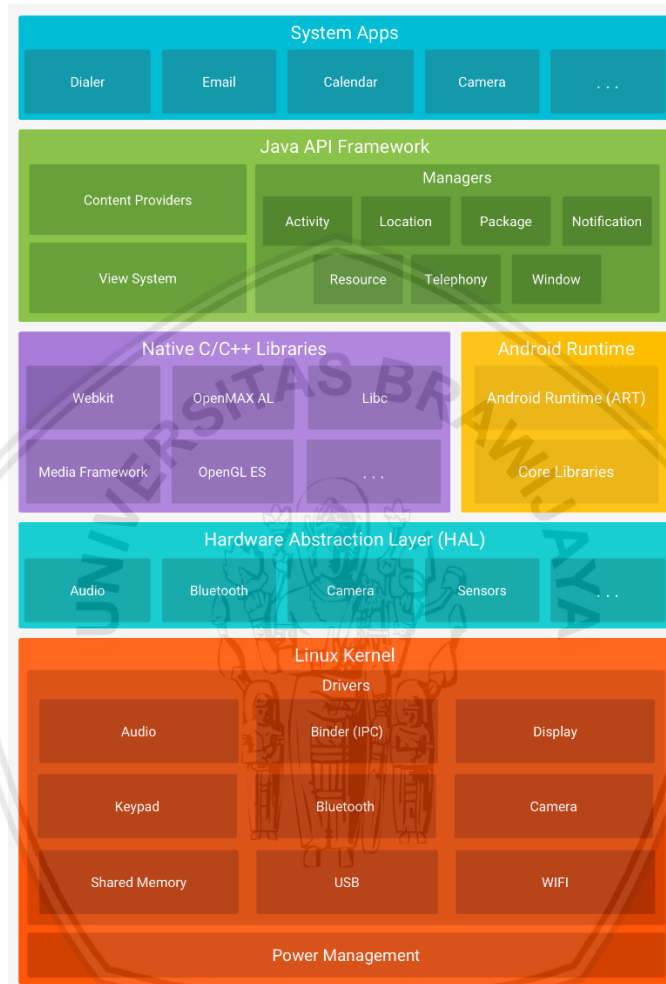
Tabel 2.1 Penjelasan Penelitian Terdahulu

No	Peneliti	Judul	Hasil
1	Bambang Tri Wahyu U, Angga Widya Anggriawan (2015)	Sistem Rekomendasi Paket Wisata Se-malang Raya Menggunakan Metode Hybrid Content Based Dan Collaborative	Sistem yang dapat memberikan rekomendasi paket wisata Se – Malang Raya dengan menggunakan Algoritme Nearest Neighbor
2	Ahmad Fali Oklilas, Sri Desy Siswanti, M Dieka Rachman (2015)	Akurasi Pembacaan GPS pada Android untuk Location Based Service (Studi Kasus: Informasi Lokasi SMA di Palembang)	Sistem yang dapat memberikan perbandingan akurasi antara GPS pada Android dengan GPS komersil (Garmin)
3	Amrah Maryam, Nadia Siddiqui, Mohammed A. Qadeer, M. Sarosh Umar (2016)	Travel Management System using GPS & Geo Tagging on Android Platform	Sistem yang dapat memberikan petunjuk arah menggunakan GPS dengan tema dari tujuan pengguna (restoran, hotel , dan lain lain)

2.2 Android

Android ialah sebuah sistem operasi pada *smartphone* berbasis *linux*. Android menyajikan *platform* terbuka kepada pengembang sistem untuk membangun aplikasi (Dharmawan,2015). Saat ini perkembangan sistem operasi android cukup pesat. Hal ini dikarenakan android ialah *platform* yang paling lengkap, baik itu sistem operasinya, aplikasi dan *tools* pengembangan, *market* aplikasi android

serta bantuan dari komunitas *open source* di dunia (Dharmawan,2015). Pada penelitian ini,penulis menggunakan sensor GPS yang telah ada pada *smartphone* android. Sensor GPS digunakan untuk mengetahui posisi pengguna serta posisi akhir rute *gowes*. Android memiliki arsitektur sendiri yang membuatnya berbeda dari sistem operasi lainnya, berikut adalah arsitektur android yang dipaparkan pada Gambar 2.1 dibawah ini :



Gambar 2.1 Arsitektur Sistem Operasi Android

Sumber : developer android (2019)

2.2.1 Linux Kernel

Android diciptakan diatas kernel *linux* (Dharmawan,2015). *Linux* sendiri adalah sistem operasi terbuka yang memiliki keunggulan pada hal pengaturan memori dan proses. Pada kernel *linux* sendiri memiliki sebagian *drivers* diantaranya adalah Audio, *Binder (IPC)*, *Display*, *Keypad*, *Bluetooth*, *Camera*, *Shared Memory*, *USB* serta *WiFi*.

2.2.2 Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer merupakan *interface* atau penghubung antara perangkat keras yang terdapat dalam *smartphone* dengan *drivers* dalam kernel *linux*. HAL terdiri dari sebagian modul pustaka, yang menerapkan *interface* untuk unsur perangkat keras tertentu seperti modul kamera atau *Bluetooth* (Developer,2019).

2.2.3 Android Runtime

Android Runtime ialah mesin *virtual* yang mengeksekusi file DEX dan berjalan pada perangkat memori rendah. Fitur utama yang ada pada *android runtime* sendiri yaitu kompilasi mendahului waktu (AOT) dan tepat waktu (JIT), pengumpulan sampah (GC) yang dimaksimalkan, serta bantuan *debug* yang lebih baik. *Android Runtime* terdiri dari *Android Runtime* (ART) dan *Core Libraries* (Developer,2019).

2.2.4 Native C/C++ Libraries

Native C/C++ Libraries adalah kumpulan pustaka bahasa C/C++ yang digunakan untuk mendukung arsitektur android. Beberapa komponen inti android seperti ART dan HAL diciptakan dengan kode asli yang membutuhkan pustaka asli yang tertulis dalam bahasa C dan C++ (Developer,2019). Perangkat android memungkinkan kerangka kerja Java API membaca dan menggunakan fungsionalitas beberapa pustaka asli pada aplikasi (Developer,2019).

2.2.5 Java API Framework

Java API Framework menyajikan kelas – kelas yang bisa dimanfaatkan untuk pembangunan aplikasi android, selain itu menyediakan juga abstrak generik untuk mengakses perangkat serta bentuk aplikasi dan sumber daya aplikasi (Dharmawan,2015). Dalam *Java API Framework* terdiri dari *Content Providers*, *View System*, *Activity*, *Location*, *Package*, *Notification*, *Resource*, *Telephony*, dan *Window*. *Content Providers* berfungsi untuk menyingkat data yang membolehkan digunakan oleh aplikasi lain contohnya yakni daftar nama. *Activity* digunakan untuk menyusun siklus hidup aplikasi serta melindungi keadaan *backstack* untuk navigasi pengguna.

2.2.6 System Apps

System Apps merupakan aplikasi yang terlihat oleh pengguna dalam menggunakan sistem operasi android. *System Apps* sendiri adalah lapisan paling atas dari arsitektur android. Beberapa contoh dari lapisan ini adalah aplikasi email, kamera, kalender dan lain lain. Saat pengguna menjalankan aplikasi kalender maka lapisan ini yang akan diakses terlebih dahulu. Namun pengguna tidak mengetahui bagaimana aplikasi ini berjalan dibelakang sistem.

2.3 GPS Smartphone

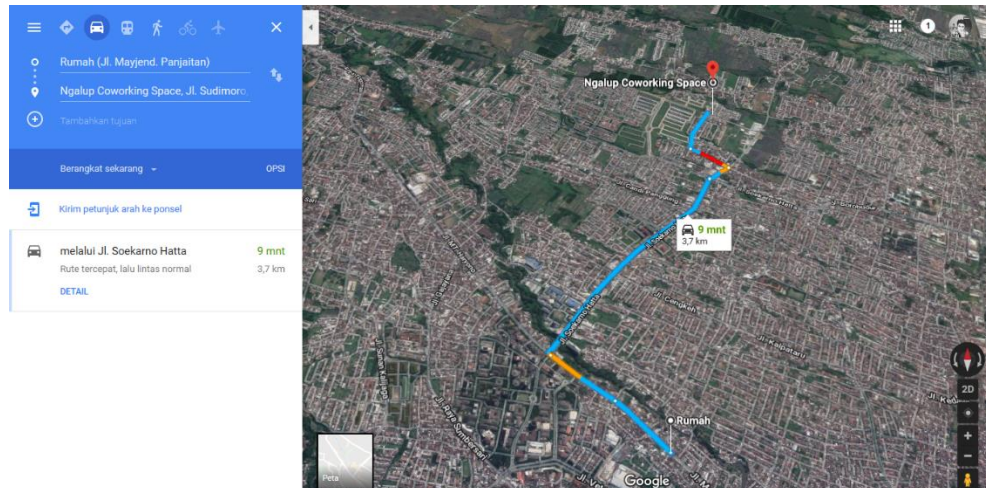
GPS (*Global Positioning System*) ialah sistem petunjuk untuk menemukan lokasi/letak dengan memanfaatkan satelit. GPS dapat memberikan lokasi/posisi suatu objek dengan cepat dan akurat serta memberikan informasi waktu serta kecepatan benda bergerak secara terus – menerus di seluruh dunia (Lengkong,2015). GPS ialah sistem yang digunakan untuk menetapkan posisi objek pada bidang bumi dengan dukungan sinyal satelit (Avelina,2018). Cara kerja GPS adalah dengan menyalurkan sinyal arus mikro ke bumi setelah itu sinyal tersebut akan ditangkap oleh perangkat penerima yang ada di bumi serta akan dimanfaatkan untuk menetapkan lokasi,kecepatan,arah dan waktu (Avelina,2018). GPS sendiri tersusun atas 30 satelit dimana 3 satelit merupakan satelit cadangan dan memiliki 6 lintasan satelit sehingga seluruh daerah di permukaan bumi dapat terpantau selama sehari penuh dalam 1 tahun (Avelina,2018). Untuk menemukan lokasi/posisi dari suatu objek,GPS menggunakan koordinat garis bujur serta garis lintang bumi melalui data yang dikirimkan oleh satelit (Maryam,2016). GPS memiliki kegunaan yang berbeda beda yaitu GPS untuk militer yang digunakan untuk membantu sistem pertahanan militer suatu negara, GPS sebagai petunjuk yang digunakan untuk memberikan petunjuk perjalanan, GPS sebagai sistem informasi geografis yang digunakan untuk pembentukan peta serta GPS untuk pemantauan gempa yang digunakan untuk memperkirakan kemungkinan terjadinya gempa di suatu wilayah (Avelina,2018). Dalam penelitian ini,GPS berperan untuk menemukan lokasi/posisi dari pengguna dan tujuan akhir rute *gowes*.

2.4 Google Maps

Google Maps ialah fasilitas denah digital milik Google yang bisa dimanfaatkan oleh pengembang aplikasi Android untuk disematkan kedalam aplikasi mereka. Google Maps secara otomatis akan mengirimkan data berupa tampilan peta serta respon pengguna berdasarkan permintaan pengguna langsung dari *server* Google Maps (Google,2019). Untuk meningkatkan pengalaman pengguna, Google Maps menyediakan beberapa fitur yang bisa dikembangkan lebih jauh seperti :

1. Mengubah penanda
2. Membuat garis petunjuk (*Polylines*)
3. Membuat peta 2D / 3D

Pada penelitian ini,peneliti menggunakan Google Maps untuk menampilkan peta sebagai petunjuk jalan / rute dari posisi pengguna ke letak alternatif (tujuan akhir rute *gowes*). Peneliti akan menggunakan data garis bujur dan lintang dari Google Maps untuk menghitung jarak rute *gowes* pengguna. Serta peneliti akan menggunakan fitur membuat garis petunjuk serta penanda dalam pengembangan aplikasi sistem ini. Contoh penggunaan Google Maps untuk rekomendasi rute perjalanan adalah seperti yang tertera pada Gambar 2.2 :



Gambar 2.2 Rekomendasi Rute dengan Google Maps

Sumber : google maps (2019)

2.5 Google Firebase

Firebase yakni fasilitas yang diberikan oleh *Google* untuk pengembang aplikasi maupun website dalam hal mengembangkan aplikasi tersebut agar lebih berkualitas, meningkatkan basis pengguna serta dapat memberikan pengaruh berupa uang. *Firebase* sendiri dapat digunakan untuk pengembangan aplikasi *Android*, *iOS*, *Website*, *C++* serta pengembangan game dengan *Unity*. *Firebase* memiliki beberapa fitur yang bisa pengguna gunakan untuk membuat aplikasi diantaranya *Cloud Firestore*, *ML Kit*, *Cloud Functions*, *Authentication*, *Hosting*, *Cloud Storage*, dan *Realtime Database*. Sementara itu jika pengguna menginginkan *Firebase* sebagai peningkatan kualitas produk dapat menggunakan fitur *Crash Reporting*, *Performance Monitoring*, dan *Test Lab*. Sedangkan jika pengguna menggunakan *Firebase* untuk keperluan bisnis dapat memanfaatkan fitur *Analytics*, *Predictions*, *Pengujian A/B Firebase*, *Cloud Messaging*, *Remote Config*, *Dynamic Links*, *App Indexing* serta *Invites* (Firebase, 2019). Dalam penelitian ini penulis memanfaatkan fitur *Realtime Database* dari *Firebase* untuk menyimpan data – data pengguna serta menyimpan data alternatif yang didapatkan dari hasil perhitungan TOPSIS kriteria yang pengguna berikan. *Firebase* memiliki logo seperti yang tertera pada Gambar 2.3 :



Gambar 2.3 Logo Google Firebase

Sumber : firebase (2019)

2.6 Metode TOPSIS

Metode TOPSIS ialah metode dalam sistem pendukung keputusan yang pertama kali diterapkan pada tahun 1981 oleh Yoon dan Hwang (Putra,2016). TOPSIS dimanfaatkan dalam menyelesaikan persoalan terkait proses pengutipan keputusan secara efisien.

2.6.1 Pengertian TOPSIS

TOPSIS (*Technique for Order Performance of Similarity to Ideal Solution*) ialah sistem pendukung keputusan multikriteria. TOPSIS memiliki cara kerja yaitu memberikan alternatif pilihan yang memiliki jarak terdekat dari solusi ideal positif serta memiliki jarak terjauh dari solusi ideal negatif pada perspektif geometris dengan mengaplikasikan jarak *Euclidean* (jarak diantara dua titik) (Windarto,2017). Dalam penggunaannya TOPSIS memiliki keunggulan yaitu konsep lugas dan ringan dipahami, kalkulasi algoritma yang praktis dan mempunyai kapabilitas untuk menilai kinerja relatif dari pilihan – pilihan keputusan dalam model matematis yang lugas serta dalam penggunaan pada *platform* Android, metode TOPSIS ini memiliki tingkat kompleksitas yang rendah (Putra,2016).

2.6.2 Langkah – langkah Perhitungan TOPSIS

Berikut merupakan langkah – langkah dalam perhitungan TOPSIS (Windarto,2017) :

- Menentukan normalisasi matriks keputusan seperti pada Persamaan 2.1 berikut :

$$R_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}; \quad (2.1)$$

Keterangan :

R_{ij} = Hasil normalisasi *decision* matriks R

$i = 1,2,...,m$

$j = 1,2,...,n$

- Menentukan bobot ternormalisasi matriks keputusan seperti pada Persamaan 2.2 berikut :

$$y_{ij} = w_{ij}r_{ij}; \quad (2.2)$$

Keterangan :

y_{ij} = Elemen matriks ternormalisasi terbobot $[i][j]$

$i = 1,2,...,m$

$j = 1,2,...,n$

- c. Menentukan nilai matriks untuk solusi ideal positif dan negatif seperti Persamaan 2.3 berikut :

$$\begin{aligned} A^+ &= (y_1^+, y_2^+, \dots, y_n^+); \\ A^- &= (y_1^-, y_2^-, \dots, y_n^-); \end{aligned} \quad (2.3)$$

Keterangan :

$$y_j^+ = \begin{cases} \max y_{ij} ; \text{jika } j \text{ adalah atribut keuntungan} \\ \min y_{ij} ; \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_i^- = \begin{cases} \min y_{ij} ; \text{jika } i \text{ adalah atribut keuntungan} \\ \max y_{ij} ; \text{jika } i \text{ adalah atribut biaya} \end{cases}$$

- d. Menghitung jarak antara alternatif A_i dengan solusi ideal positif seperti Persamaan 2.4 berikut :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}; \quad (2.4)$$

Keterangan :

D_i^+ = Jarak alternatif ke-i dengan solusi ideal positif

$i = 1, 2, \dots, m$

- e. Menghitung jarak antara alternatif A_i dengan solusi ideal negatif seperti Persamaan 2.5 berikut :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2}; \quad (2.5)$$

Keterangan :

D_i^- = Jarak alternatif ke-i dengan solusi ideal negatif

$i = 1, 2, \dots, m$

- f. Menentukan nilai prefensi untuk setiap alternatif (V_i) seperti Persamaan 2.6 berikut :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; \quad (2.6)$$

Keterangan :

V_i = Kedekatan tiap alternatif terhadap solusi ideal

$i = 1, 2, \dots, m$

2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak ialah suatu prosedur atau sebaris mekanisme yang disusun untuk suatu sistem maupun program telah cocok dan berjalan secara semestinya dengan acuan kebutuhan sistem telah terpenuhi (Karuniawati, 2015).

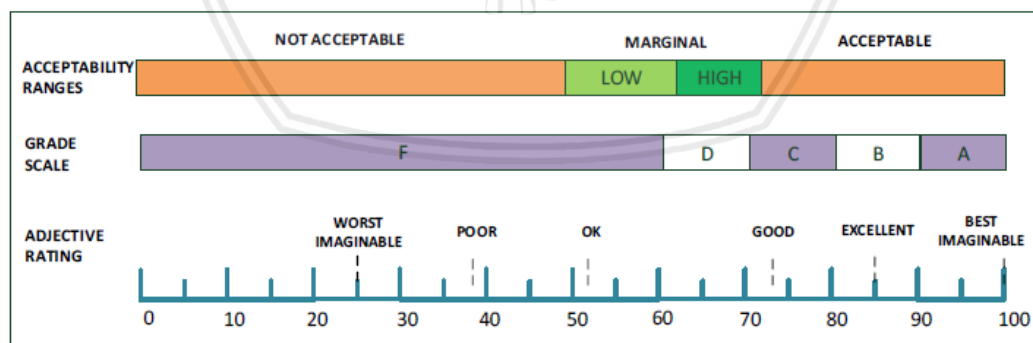
2.7.1 Pengujian Blackbox

Pengujian black box ialah mekanisme pengujian tanpa perlu mengetahui struktur internal dari suatu software / sistem karena teknik pengujian ini hanya berfokus pada masukan dan keluaran dari sistem terhadap spesifikasi suatu software (Karuniawati,2015). Beberapa kesalahan yang berusaha ditemukan dalam teknik pengujian black box ini adalah sebagai berikut (Mimin,2016) :

1. Fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan struktur data
4. Kesalahan kinerja program
5. Kesalahan terminasi

2.7.2 Pengujian Usability

Pengujian *Usability* memiliki dua metode dalam penerapannya yaitu dengan metode *heuristic evaluation (HE)* dan *system usability scale (SUS)* (Ependi,2017). *Heuristic evaluation (HE)* merupakan pengujian *usability* yang dilakukan oleh para ahli atau pakar sedangkan *system usability scale (SUS)* ialah pengujian *usability* yang dilakukan oleh pengguna akhir (Ependi,2017). Namun kedua metode tersebut memiliki kesamaan yaitu berfokus pada penilaian interaksi antarmuka sistem perangkat lunak (Ependi,2017). Pada metode *system usability scale (SUS)* menggunakan 10 pernyataan sebagai alat pengujian, metode ini juga tidak perlu menggunakan banyak sampel sehingga dapat meminimalisir biaya dalam pengujian (Ependi,2017). Hasil akhir dari pengujian *usability* dengan metode *system usability scale (SUS)* adalah *grade* nilai yang ada pada Gambar 2.4 :

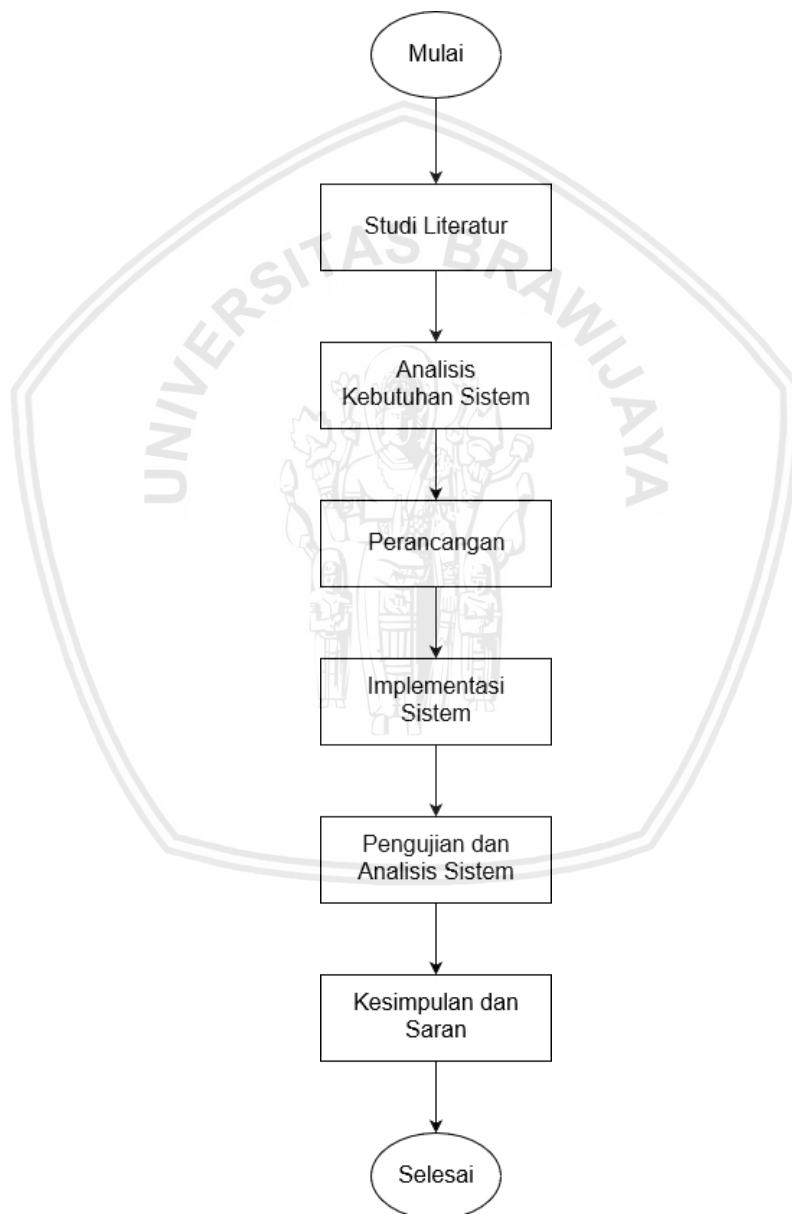


Gambar 2.4 Penentuan Hasil Penilaian System Usability Scale (SUS)

Sumber : Ependi, Panjaitan , dan Hutrianto (2017)

BAB 3 METODOLOGI

Bab metodologi menjabarkan metode serta langkah – langkah yang akan digunakan pada penelitian ini. Tahapan yang akan dilalui dalam penelitian adalah studi literatur, pengumpulan data, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, dan kesimpulan serta saran. Sedangkan untuk metode penelitian sendiri, penulis memilih metode wawancara dalam proses pengambilan data penelitian. Langkah – langkah penelitian dijabarkan pada Gambar 3.1 dibawah ini :



Gambar 3.1 Diagram Alir Metodologi

3.1 Studi Literatur

Studi literatur memiliki tujuan untuk mempelajari, menekuni, serta menjabarkan teori – teori maupun konsep yang diterapkan dalam penelitian. Teori dan konsep yang dikumpulkan oleh penulis merupakan teori dan konsep dari penelitian terdahulu yang mendapat kesamaan topik dengan penulis. Teori dan konsep didapatkan dari buku, jurnal, maupun artikel yang memiliki hubungan dengan topik penelitian penulis. Berikut merupakan studi literatur yang diterapkan dalam penelitian ini :

1. Kajian pustaka tentang penelitian terdahulu
2. Android
3. GPS Smartphone
4. Google Maps SDK
5. Firebase
6. Metode TOPSIS
7. Pengujian Perangkat Lunak

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem ialah langkah metodologi penelitian yang bertujuan untuk memperoleh data kebutuhan yang hendak digunakan dalam sistem. Analisis kebutuhan sistem memiliki beberapa faktor didalamnya yaitu deskripsi umum sistem, kebutuhan *fungsi* sistem, kebutuhan *non-fungsi* sistem, serta identifikasi pengguna sistem. Kebutuhan *fungsi* serta *non-fungsi* sistem akan dijelaskan menggunakan *usecase diagram* dan *usecase skenario*. Dalam penelitian ini, penulis menggunakan teknik wawancara langsung dengan memberikan beberapa pertanyaan seputar olahraga sepeda dikota malang sehingga nantinya penulis mendapatkan data aktifitas bersepeda seperti waktu tempuh, jarak, rute dan lain sebagainya. Wawancara langsung akan berfokus kepada responden yang memiliki keterkaitan dengan bersepeda. Lalu dalam mengolah data tersebut, penulis menggunakan *tools* Microsoft Word untuk mengolah hasil dari wawancara dan akan melakukan perhitungan manual dengan metode TOPSIS pada *tools* Microsoft Excel.

3.3 Perancangan

Perancangan adalah tahap untuk merubah kebutuhan menjadi spesifikasi terperinci (Lemke,2018). Perancangan ini meliputi perancangan antarmuka, perancangan kebutuhan yang harus ada didalam sistem, serta perancangan basis data (Lemke,2018).

3.3.1 Perancangan Sistem

Perancangan sistem ialah langkah metodologi penelitian yang dilakukan ketika analisis kebutuhan sistem telah terpenuhi. Dalam perancangan sistem terdapat

beberapa langkah – langkah yaitu perencanaan antarmuka sistem, perencanaan *class diagram*, perencanaan *sequence diagram*, perencanaan *activity diagram*, serta perencanaan basis data. Pada penelitian ini, penulis menggunakan dua *tools* untuk membantu dalam merancang sistem yaitu Adobe XD untuk merancang antarmuka sistem dan Visual Paradigm untuk merancang *class diagram*, *sequence diagram*, *activity diagram* serta merancang basis data.

3.3.2 Perancangan Algoritme

Perancangan algoritme sistem sendiri menggunakan metode TOPSIS. Dalam metode ini terdapat perhitungan manual yang nantinya akan menggunakan bantuan *tools* Microsoft Excel untuk mendapatkan hasil yang diinginkan. Metode TOPSIS ini akan digunakan untuk melakukan perankingan terhadap data uji untuk mendapatkan rekomendasi alternatif atas rute *gowes* di kota Malang.

3.4 Implementasi Sistem

Implementasi sistem ialah langkah metodologi penelitian yang menjadi point utama pengembangan sistem. Dalam implementasi sistem mencakup batasan implementasi sistem, spesifikasi sistem, implementasi antarmuka sistem, implementasi class, implementasi basis data serta implementasi kode program dalam sistem.

Untuk implementasi *interface* sistem menggunakan *smartphone* dengan *operating system Android*, implementasi basis data sistem memanfaatkan *Google Firebase*, implementasi class serta kode program menggunakan *software Android Studio* dengan bahasa pemrograman *Java*.

3.5 Analisis dan Pengujian Sistem

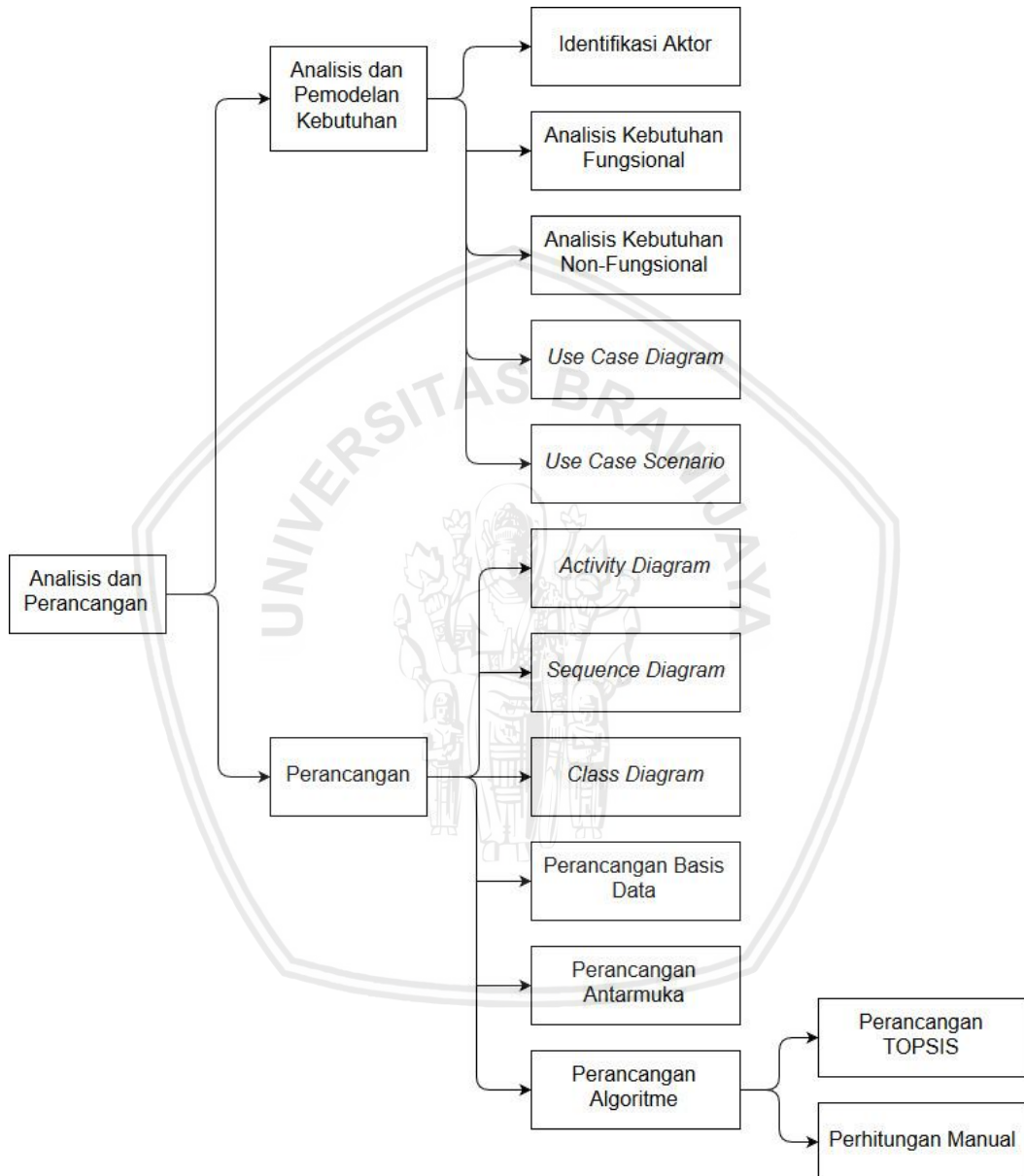
Setelah sistem selesai dikembangkan maka langkah selanjutnya yaitu menganalisa serta melakukan pengujian sistem. Analisis pada penelitian ini berupa analisis hasil dari perhitungan manual dengan perhitungan sistem. Selanjutnya untuk memvalidasi apakah kebutuhan *fungsi* yang telah ditulis berjalan dengan semestinya adalah dengan menggunakan pengujian *blackbox*. Pengujian *Usability* digunakan untuk mendapatkan hasil penilaian dari pengguna akhir sistem rekomendasi rute *gowes* di Kota Malang. Dalam pengujian *Usability* sendiri diterapkan metode *system usability scale (SUS)*.

3.6 Kesimpulan dan Saran

Kesimpulan dan saran dilakukan sesudah tahap analisis dan pengujian sistem selesai dilakukan. Kesimpulan dilakukan untuk menjawab permasalahan yang telah didefinisikan pada rumusan masalah sedangkan saran dilakukan untuk memberikan informasi kepada peneliti selanjutnya untuk meningkatkan sistem ini agar lebih baik.

BAB 4 ANALISIS DAN PERANCANGAN

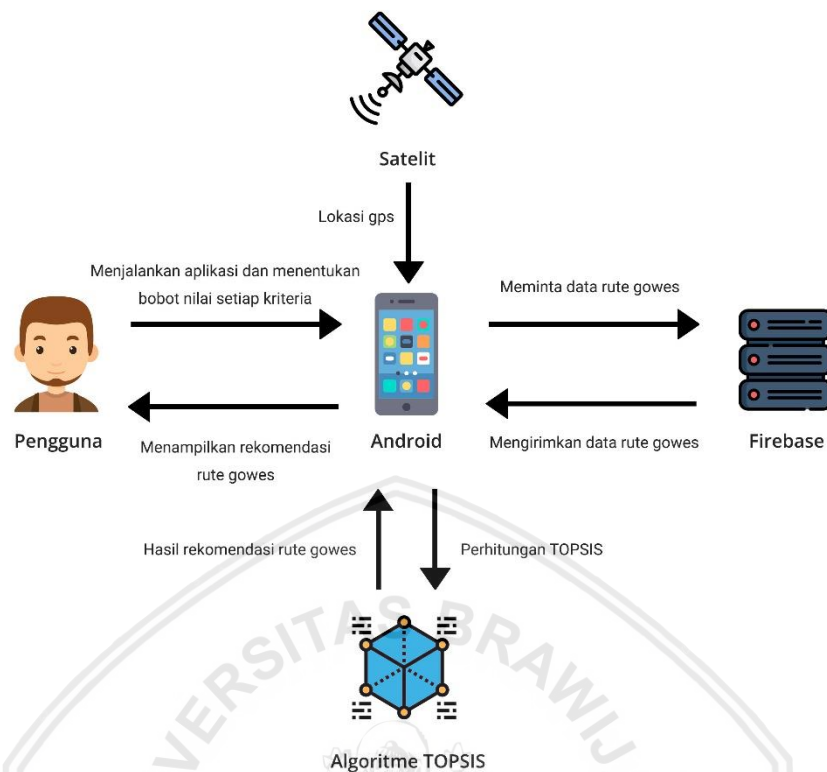
Bab analisis dan perancangan menerangkan tentang analisis dan perancangan dari aplikasi rekomendasi rute gowes di Kota Malang. Langkah – langkah analisis dan perancangan dijabarkan pada Gambar 4.1 dibawah ini :



Gambar 4.1 Diagram Alir Analisis dan Perancangan

Gambaran Aplikasi

Gambaran aplikasi menjelaskan alur kerja sistem rekomendasi rute gowes secara umum. Gambaran aplikasi ditunjukkan pada Gambar 4.2 :



Gambar 4.2 Gambaran Aplikasi

Pengguna mengaktifkan sistem rekomendasi rute gowes pada perangkat Android. Selanjutnya pengguna memberikan bobot nilai untuk masing – masing kriteria (jarak rute gowes, waktu tempuh, dan kesulitan medan). Saat pengguna melakukan pengisian bobot nilai, aplikasi secara otomatis akan mengambil data lokasi GPS perangkat Android pengguna. Selanjutnya *firebase* akan memberikan data rute gowes yang ada di kota Malang. Aplikasi akan menerima semua data tersebut dan akan melakukan perhitungan menggunakan algoritme TOPSIS. Hasil perhitungan TOPSIS akan ditampilkan kembali kepada pengguna. Pengguna memperoleh anjuran yang diinginkan berlandaskan bobot nilai yang diberikan.

4.1 Analisis Kebutuhan

Dalam tahap analisis kebutuhan ini dijabarkan terkait kebutuhan – kebutuhan yang akan mendukung sistem rekomendasi rute gowes di kota Malang. Didalamnya terdapat gambaran aplikasi, identifikasi aktor, analisis kebutuhan fungsional sistem, analisis kebutuhan non-fungsional sistem, *use case diagram* serta *use case scenario*. Analisis kebutuhan ini menjelaskan apa saja yang berada di dalam sistem rekomendasi rute gowes. Dalam analisis kebutuhan terdapat kode yang menunjukan suatu kebutuhan termasuk kebutuhan fungsional atau non – fungsional. Penjelasan kode tersebut terdapat pada Gambar 4.3 :



Gambar 4.3 Penjelasan Kode Kebutuhan Sistem

4.1.1 Identifikasi Aktor

Langkah ini menjabarkan tokoh atau seseorang yang berhubungan dengan sistem rekomendasi rute gowes. Penjelasan terkait aktor dijabarkan pada Tabel 4.1 dibawah ini :

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna mampu memberikan bobot nilai untuk masing – masing kriteria yaitu jarak rute gowes, waktu tempuh dan kesulitan medan. Selain itu pengguna mampu menilik hasil rekomendasi berbentuk list serta dapat melihat rincian dari list tersebut. Pengguna dapat melihat rute gowes menggunakan <i>Google Maps</i>

4.1.2 Analisis Kebutuhan Fungsional

Sub bab ini berisikan kebutuhan fungsional yang akan digunakan pada perancangan sistem rekomendasi rute gowes. Kebutuhan fungsional ini ialah kebutuhan dari aktor pengguna. Kebutuhan fungsional untuk sistem rekomendasi rute gowes ini akan dipaparkan pada Tabel 4.2 dibawah ini :

Tabel 4.2 Kebutuhan Fungsional Sistem

Kode	Nama <i>Use Case</i>	Kebutuhan Fungsional
RRG – F – 01	Memberikan bobot nilai untuk setiap kriteria	Sistem mampu memperoleh masukan pengguna berbentuk bobot nilai setiap kriteria
RRG – F – 02	Melihat rekomendasi rute gowes	Sistem mampu menampilkan rekomendasi rute gowes sesuai bobot nilai yang dimasukan pengguna

RRG – F – 03	Melihat detail rekomendasi rute	Sistem mampu menampilkan detail dari rekomendasi rute gowes
RRG – F – 04	Melihat petunjuk arah	Sistem mampu menunjukan arah rute gowes dengan bantuan aplikasi <i>Google Maps</i>

Kode RRG – F – 01 menjelaskan kebutuhan fungsional untuk menerima masukan pengguna berupa bobot nilai masing – masing kriteria. Sebelumnya pengguna menekan tombol mulai gowes sehingga pengguna aplikasi akan menampilkan halaman bobot nilai. Setelah bobot nilai dimasukan sesuai dengan keinginan pengguna dan pengguna menekan tombol lihat rute gowes maka aplikasi akan menjalankan perhitungan TOPSIS dan menampilkan hasil rekomendasi rute gowes.

Kode RRG – F – 02 menjelaskan kebutuhan fungsional untuk melihat rekomendasi rute gowes. Setelah menekan tombol lihat rute gowes maka pengguna akan dialihkan ke halaman selanjutnya yaitu halaman rekomendasi rute gowes. Pada halaman ini, pengguna akan melihat rute gowes mana yang sesuai dengan bobot nilai yang dimasukan sebelumnya berupa list.

Kode RRG – F – 03 menjelaskan kebutuhan fungsional untuk melihat detail rekomendasi rute gowes. Pada halaman ini, pengguna dapat melihat detail tentang rute gowes berupa nama rute, jarak tempuh, waktu tempuh, dan tingkat kesulitan medan. Pada halaman ini juga termuat 2 tombol yaitu tombol untuk melihat rute gowes dan tombol untuk menuju ke lokasi rute gowes.

Kode RRG – F – 04 menjelaskan kebutuhan fungsional untuk melihat petunjuk arah rute gowes. Pada halaman detail rekomendasi terdapat 2 tombol yaitu tombol untuk melihat rute gowes dan tombol untuk menuju ke lokasi rute gowes. Jika pengguna menekan tombol melihat rute gowes maka aplikasi secara otomatis akan mengarahkan ke Google Maps dengan menampilkan titik mulai dan titik berakhirnya rute gowes sedangkan tombol untuk menuju ke lokasi akan menampilkan lokasi pengguna terkini menuju titik awal rute gowes.

4.1.3 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ialah kebutuhan sistem yang tidak diharapkan secara lisan oleh pengguna tetapi menunjang penggunaan sistem itu sendiri. Kebutuhan non-fungsional untuk sistem rekomendasi rute gowes ini akan dipaparkan pada Tabel 4.3:

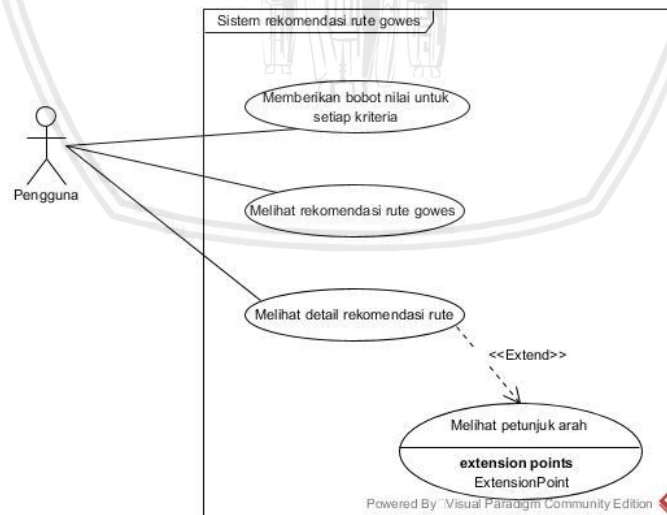
Tabel 4.3 Kebutuhan Non-Fungsional Sistem

Kode	Nama <i>Use Case</i>	Deskripsi
RRG – NF – 01	Validasi Algoritme	Melakukan validasi hasil keluaran sistem dengan

		kalkulasi manual. Penggunaan metode TOPSIS dikatakan berhasil ketika kalkulasi manual sama dengan hasil keluaran sistem
--	--	---

4.1.4 Use Case Diagram

Sistem rekomendasi rute gowes akan berjalan ketika pengguna menekan icon aplikasi pada *smartphone* android. Kemudian aplikasi akan menampilkan halaman pembuka dengan logo dan tombol mulai gowes. Pengguna menekan tombol mulai gowes dan aplikasi akan menyajikan halaman baru dengan beberapa parameter untuk diisi. Parameter tersebut yaitu jarak rute gowes, waktu tempuh dan kesulitan medan yang bisa diisi oleh user dengan nilai antara 0 – 1. Setelah pengguna menekan tombol lihat rute gowes maka aplikasi akan menyajikan anjuran rute gowes yang selaras dengan bobot nilai yang dimasukan oleh pengguna pada halaman sebelumnya. Pengguna mampu melihat detail rute gowes dengan menentukan salah satu list anjuran yang ada. Pada halaman detail rute gowes terdapat rincian tentang rute gowes yang dipilih beserta 2 tombol petunjuk arah. Dua tombol tersebut yaitu tombol untuk melihat rute gowes dan tombol untuk menuju ke lokasi rute gowes. Jika pengguna menekan tombol melihat rute gowes maka aplikasi secara otomatis akan mengarahkan ke Google Maps dengan menampilkan titik mulai dan titik berakhirnya rute gowes sedangkan tombol untuk menuju ke lokasi akan menampilkan lokasi pengguna terkini menuju titik awal rute gowes. Use case diagram untuk tokoh pengguna ini dapat dilihat pada Gambar 4.4:



Gambar 4.4 Use Case Diagram Sistem Rekomendasi rute Gowes

4.1.5 Use Case Scenario

Use case scenario ialah penjabaran rinci dari *use case diagram* yang sebelumnya anda lihat. *Use case scenario* sistem rekomendasi rute gowes ini

tersusun dari nama *use case diagram*, nama aktor yang terlibat dalam kebutuhan tersebut, deskripsi singkat terkait kebutuhan tersebut, kondisi awal sistem, alur sistem, alur alternatif, dan kondisi akhir yang diharapkan dari sistem.

4.1.5.1 Memberikan Bobot Nilai Untuk Setiap Kriteria

Memberikan bobot nilai untuk setiap kriteria adalah *use case scenario* dimana pengguna memberikan bobot nilai untuk kriteria jarak rute gowes, waktu tempuh dan kesulitan medan. Detail terkait *use case scenario* ini akan di jelaskan pada Tabel 4.4 :

Tabel 4.4 Memberikan Bobot Nilai Untuk Setiap Kriteria

Nama	Memberikan bobot nilai untuk setiap kriteria
Aktor	Pengguna
Deskripsi	Pada <i>use case</i> ini pengguna mampu memberikan bobot untuk setiap kriteria sesuai keinginan pengguna
Pre-condition	Pengguna sudah membuka aplikasi serta menekan tombol mulai gowes
Main flow	1. Aplikasi menyajikan halaman bobot nilai 2. Pengguna memberikan bobot nilai pada masing – masing kriteria 3. Pengguna menekan tombol lihat rute gowes
Alternative flow	-
Post-condition	Aplikasi mengarahkan ke halaman hasil rekomendasi rute gowes

4.1.5.2 Melihat Rekomendasi Rute Gowes

Melihat rekomendasi rute gowes adalah *use case scenario* dimana pengguna mampu menyajikan rekomendasi rute gowes sesuai dengan bobot yang telah diberikan sebelumnya. Detail terkait *use case scenario* ini akan di jelaskan pada Tabel 4.5 :

Tabel 4.5 Melihat Rekomendasi Rute Gowes

Nama	Melihat rekomendasi rute gowes
Aktor	Pengguna
Deskripsi	Pada <i>use case</i> ini pengguna mampu melihat rekomendasi rute gowes sesuai dengan bobot yang telah diberikan sebelumnya
Pre-condition	Pengguna menekan tombol lihat rute gowes

Main flow	1. Aplikasi menyajikan hasil rekomendasi rute gowes 2. Pengguna menentukan salah satu hasil rekomendasi rute gowes
Alternative flow	-
Post-condition	Aplikasi mengarahkan ke halaman detail rute gowes

4.1.5.3 Melihat Detail Rekomendasi Rute

Melihat detail rekomendasi rute adalah *use case scenario* dimana pengguna mampu melihat detail rekomendasi rute berupa jarak rute gowes, waktu tempuh, dan kesulitan medan dalam rute tersebut. Detail terkait *use case scenario* ini akan di jelaskan pada Tabel 4.6 :

Tabel 4.6 Melihat Detail Rekomendasi Rute

Nama	Melihat detail rekomendasi rute
Aktor	Pengguna
Deskripsi	Pada <i>use case</i> ini pengguna mampu melihat detail rekomendasi rute yang dipilih sebelumnya
Pre-condition	Pengguna memilih salah satu rekomendasi rute gowes
Main flow	1. Aplikasi menyajikan detail rekomendasi rute gowes 2. Pengguna menekan tombol melihat rute gowes dan tombol menuju lokasi.
Alternative flow	-
Post-condition	Aplikasi membuka aplikasi Google Maps

4.1.5.4 Melihat Petunjuk Arah

Melihat petunjuk arah ialah *use case scenario* dimana pengguna mampu melihat petunjuk arah ke lokasi gowes serta melihat rute gowes. Detail terkait *use case scenario* ini akan di jelaskan pada Tabel 4.7 :

Tabel 4.7 Melihat Petunjuk Arah

Nama	Melihat petunjuk arah
Aktor	Pengguna
Deskripsi	Pada <i>use case</i> ini pengguna mampu melihat petunjuk arah ke lokasi gowes dan melihat rute gowes dari titik awal ke titik akhir
Pre-condition	Pengguna berada pada halaman detail rekomendasi rute

Main flow	1. Pengguna memilih tombol lihat lokasi gowes 2. Aplikasi menyajikan arah dari lokasi pengguna saat ini ke titik awal rute gowes pada Google Maps
Alternative flow	1. Pengguna memilih tombol lihat rute gowes 2. Aplikasi menyajikan rute gowes dari titik awal sampai titik akhir rute pada Google Maps
Post-condition	Pengguna mendapatkan petunjuk arah

4.2 Perancangan

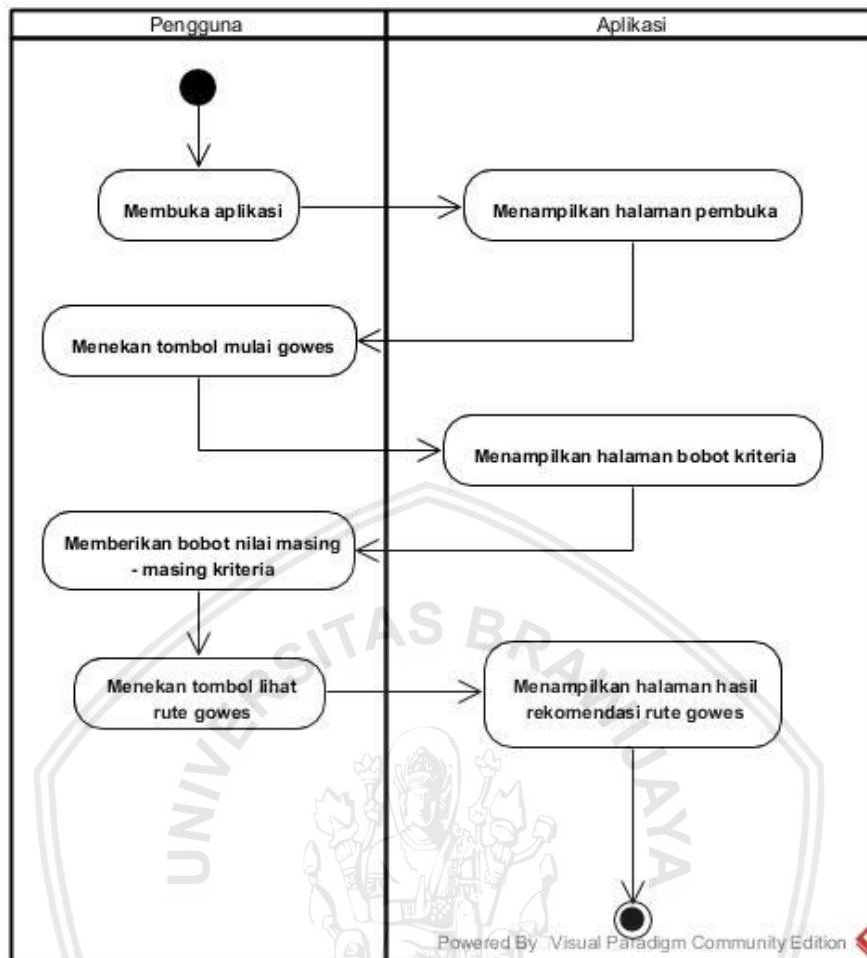
Perancangan sistem merupakan sub bab yang membahas terkait perancangan sistem pada perangkat bergerak android. Perancangan sistem sendiri tersusun atas perancangan *activity diagram*, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, perancangan antarmuka aplikasi, perancangan algoritme.

4.2.1 Perancangan Activity Diagram

Perancangan *activity diagram* ialah perancangan sistem untuk memodelkan aktivitas – aktivitas didalam sistem yang mampu dilakukan oleh pengguna. Perancangan *activity diagram* berdasarkan *use case diagram* yang dipaparkan sebelumnya.

4.2.1.1 Activity Diagram Memberikan Bobot Nilai Untuk Setiap Kriteria

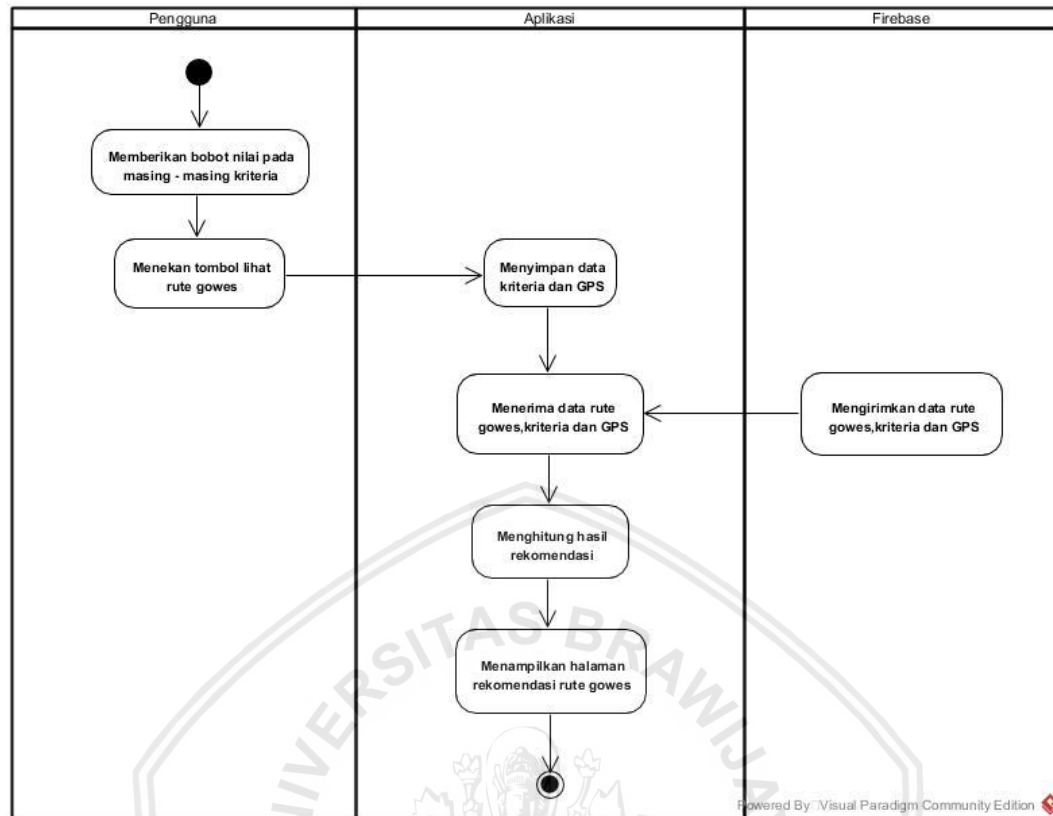
Activity diagram ini menggambarkan proses pengguna dalam melakukan pengisian bobot nilai untuk setiap kriteria. Dimulai dari pengguna menjalankan aplikasi, aplikasi akan menyajikan halaman pembuka. Pengguna menekan tombol mulai gowes. Selanjutnya aplikasi akan menyajikan halaman kriteria dimana pengguna dapat menentukan sendiri bobot nilai untuk setiap kriteria yang ada. Dalam penentuan bobot nilai untuk setiap kriteria, pengguna cukup menggeser geser saja *point* yang ada di setiap kriteria dan aplikasi nantinya akan otomatis mendapatkan nilai yang dimasukan pengguna. Kriteria untuk aplikasi ini sendiri terdapat 3 kriteria yaitu jarak rute, waktu tempuh serta kesulitan medan dari rute yang ada. Setelah pengguna memberikan nilai untuk masing – masing kriteria maka akan ditampilkan nilai dari masukan pengguna tersebut di dalam tampilan halaman kriteria sehingga pengguna bisa mengetahui berapa nilai yang telah mereka masukan sebelumnya. Selanjutnya diakhiri dengan menekan tombol lihat rute gowes. Penjelasan lengkap untuk *activity diagram* ini dapat dilihat pada Gambar 4.5 :



Gambar 4.5 Activity Diagram Memberikan Bobot Nilai Untuk Setiap Kriteria

4.2.1.2 Activity Diagram Melihat Rekomendasi Rute Gowes

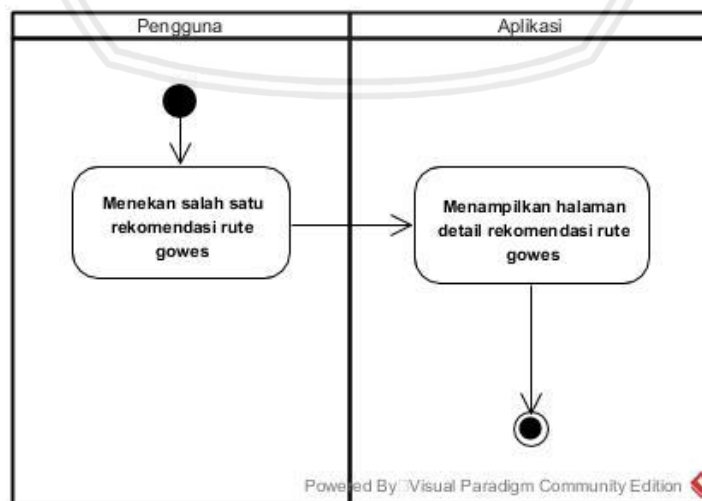
Activity diagram ini menggambarkan proses pengguna dalam melihat hasil rekomendasi rute gowes. Dimulai dengan pengguna melakukan pengisian bobot nilai yang kemudian dilanjutkan dengan menekan tombol lihat rute gowes. Pengguna harus memasukkan bobot nilai yang berjumlah tiga dari tiap – tiap kriteria. Ada pun kriteria yang terdapat dalam aplikasi ini yaitu jarak rute, waktu tempuh rute serta kesulitan medan dari rute yang ada. Aplikasi kemudian mengambil lokasi GPS pengguna, jika pengguna belum mengaktifkan GPS pada perangkat android maka aplikasi akan menampilkan pesan error. Dalam *firebase* akan mengirimkan data rute gowes. Selanjutnya aplikasi hendak mengolah data tersebut dengan menerapkan algoritme TOPSIS dan menyajikan hasil rekomendasi rute tersebut kepada pengguna .Penjelasan lengkap untuk *activity diagram* ini dapat dilihat pada Gambar 4.6 :



Gambar 4.6 Activity Diagram Melihat Rekomendasi Rute Gowes

4.2.1.3 Activity Diagram Melihat Detail Rekomendasi Rute

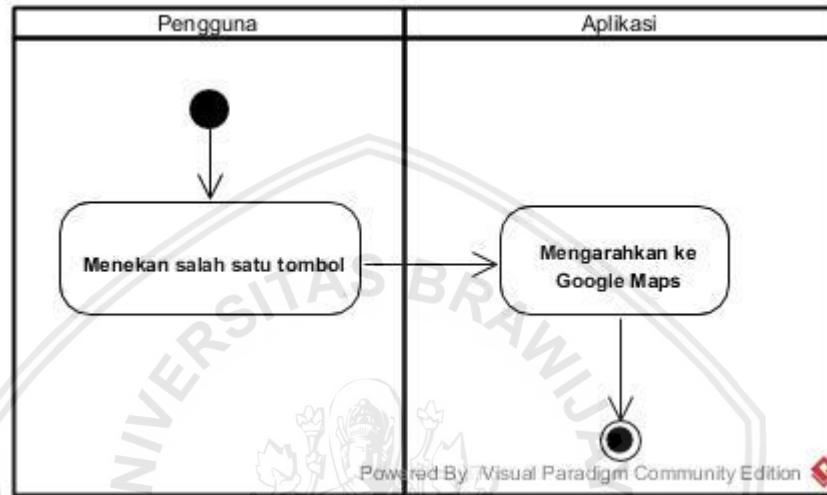
Activity diagram ini menggambarkan proses pengguna dalam melihat detail rekomendasi rute gowes. Diawali dengan pengguna menentukan salah satu rekomendasi rute gowes yang terdapat pada list. Selanjutnya aplikasi akan menampilkan detail terkait rute gowes tersebut .Penjelasan lengkap untuk *activity diagram* ini dapat dilihat pada Gambar 4.7 :



Gambar 4.7 Activity Diagram Melihat Detail Rekomendasi Rute

4.2.1.4 Activity Diagram Melihat Petunjuk Arah

Activity diagram ini menjabarkan proses pengguna dalam menemukan petunjuk arah. Pengguna mampu menentukan sendiri petunjuk arah apa yang ingin dilihat. Jika pengguna ingin melihat rute gowes dari titik awal hingga akhir maka menekan tombol lihat rute gowes. Sedangkan jika ingin melihat petunjuk dari letak saat ini ke titik awal rute bisa memilih tombol lihat lokasi gowes. Penjelasan lengkap untuk *activity diagram* ini dapat dilihat pada Gambar 4.8 dibawah ini :



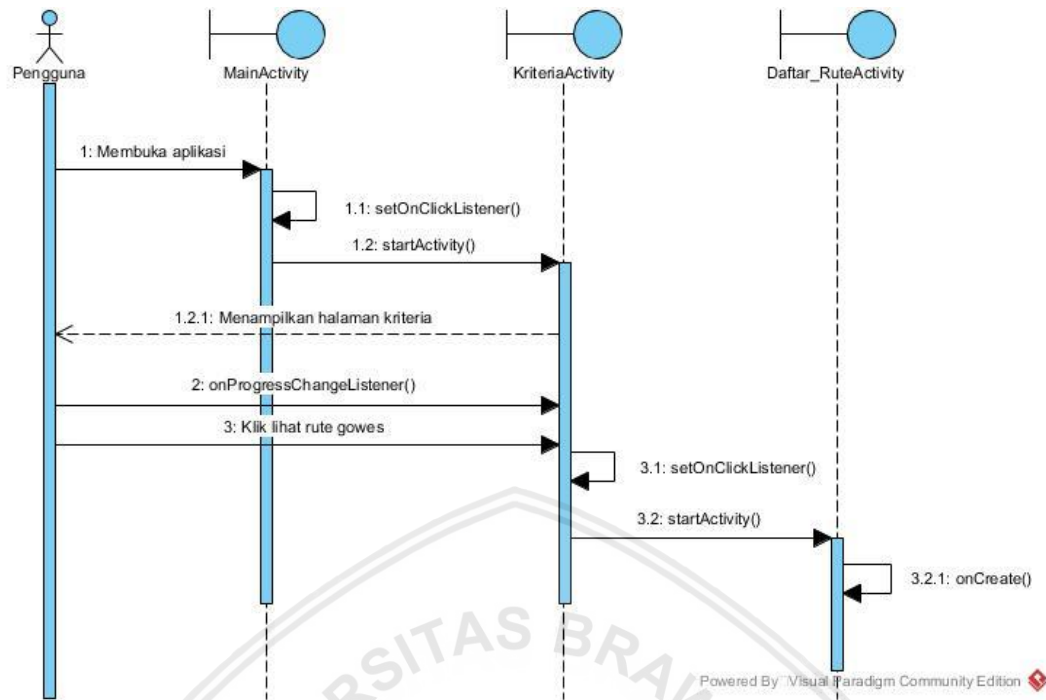
Gambar 4.8 Activity Diagram Melihat Petunjuk Arah

4.2.2 Perancangan Sequence Diagram

Sequence diagram adalah diagram yang menerangkan tentang relasi atau interaksi antara objek – objek atau *class – class* yang berada di dalam sistem rekomendasi rute gowes.

4.2.2.1 Sequence Diagram Memberikan Bobot Nilai Untuk Setiap Kriteria

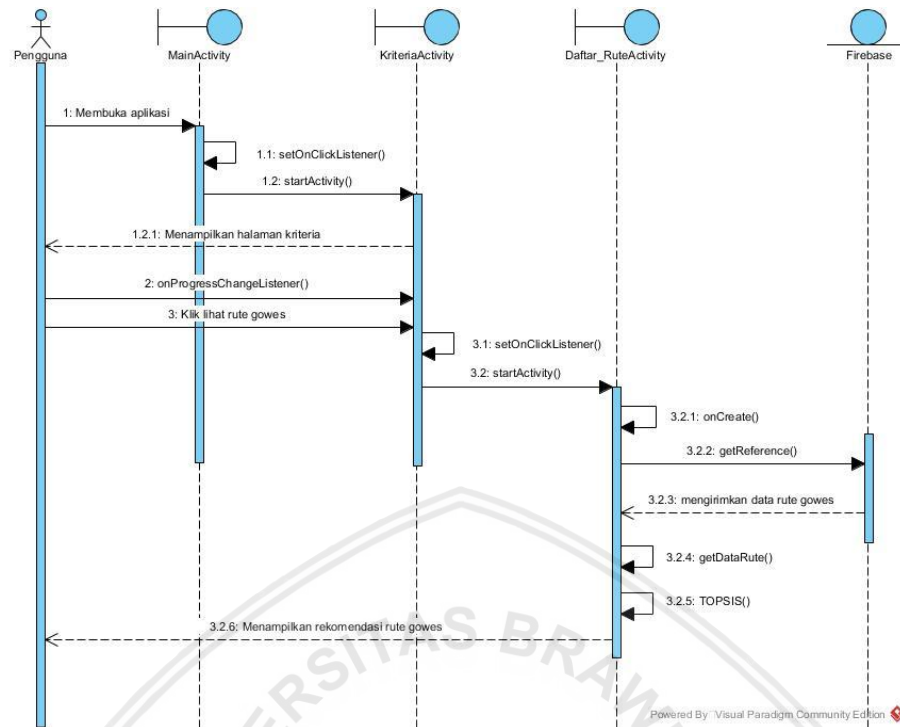
Pengguna pertama kali membuka aplikasi sistem rekomendasi rute gowes. Lalu aplikasi memanggil *MainActivity()*. Pengguna selanjutnya menekan tombol mulai gowes dan memanggil *method setOnClickListener()*. Setelah *method* dipanggil kemudian memulai *activity* baru dengan menjalankan *method startActivity()* pada *KriteriaActivity()*. Pada *activity* ini pengguna mengubah nilai dari masing – masing kriteria dan menjalankan *method onProgressChangeListener()* dan diakhiri dengan menekan tombol lihat rute gowes. Selanjutnya di jalankan *method*. Terakhir dijalankan *method startActivity()* menuju *Daftar_RuteActivity()*. Untuk penjelasan lengkapnya, digambarkan pada Gambar 4.9 :



Gambar 4.9 Sequence Diagram Memberikan Bobot Nilai Untuk Setiap Kriteria

4.2.2.2 Sequence Diagram Melihat Rekomendasi Rute Gowes

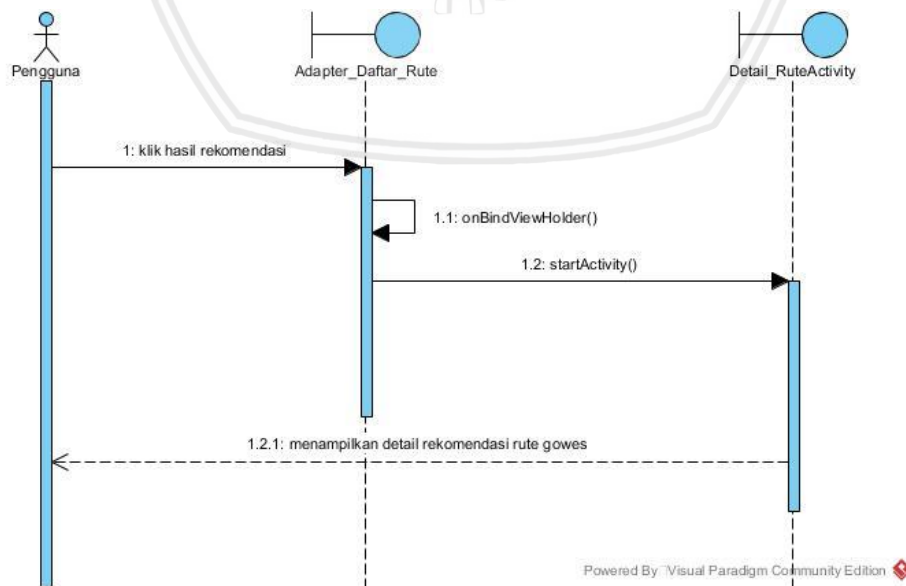
Sequence ini merupakan lanjutan dari *sequence* sebelumnya dimana *activity* yang sedang berjalan adalah *Daftar_RuteActivity()*. Pada *Daftar_RuteActivity()* ini berjalan satu *method* yaitu *method* *getReference()* dimana *method* ini bertujuan untuk menghubungkan aplikasi dengan *Firestore* sehingga nantinya data rute yang ada pada basis data bisa didapatkan oleh aplikasi. Setelah *method* *getReference()* dijalankan maka *Firestore* akan mengirimkan data rute gowes dengan *format* pengiriman data JSON. Untuk menangani *format* JSON tersebut, pada *activity* ini dipanggil sebuah *method* yaitu *method* *getData()*. Pada *method* ini nantinya data – data yang telah diterima akan dimasukkan kedalam *arraylist* dua dimensi dan hendak diterapkan untuk kalkulasi TOPSIS. Selanjutnya ialah tahap memanggil *method* TOPSIS(). Pada *method* ini semua perhitungan akan dijalankan, mulai dari perhitungan normalisasi alternatif data, perhitungan normalisasi terbobot, kalkulasi solusi ideal positif dan ideal negatif, kalkulasi jarak solusi ideal positif dan ideal negatif serta kalkulasi nilai prefensi dan perankingan. Setelah sistem selesai melakukan perhitungan TOPSIS pada *method* TOPSIS() maka aplikasi akan menampilkan hasil perhitungannya berupa daftar rute gowes yang ada di kota Malang dengan nilai prefensi tertinggi sampai dengan nilai prefensi terendah. Untuk tahap – tahap perhitungan TOPSIS secara rinci akan dijelaskan pada perhitungan manual. Untuk penjelasan lengkapnya, digambarkan pada Gambar 4.10 :



Gambar 4.10 Sequence Diagram Melihat Rekomendasi Rute Gowes

4.2.2.3 Sequence Diagram Melihat Detail Rekomendasi Rute

Ketika pengguna hendak melihat detail salah satu rekomendasi rute gowes maka pertama – tama pengguna harus memilih salah satu hasil rekomendasi. Selanjutnya aplikasi akan memanggil Adapter_Daftar_Rute() dan menjalankan *method* onBindViewHolder() untuk memanggil *method* startActivity(). Setelah *method* dijalankan maka aplikasi akan menampilkan Detail_RuteActivity() kepada pengguna. Untuk penjelasan lengkapnya, digambarkan pada Gambar 4.11 :



Gambar 4.11 Sequence Diagram Melihat Detail Rekomendasi Rute

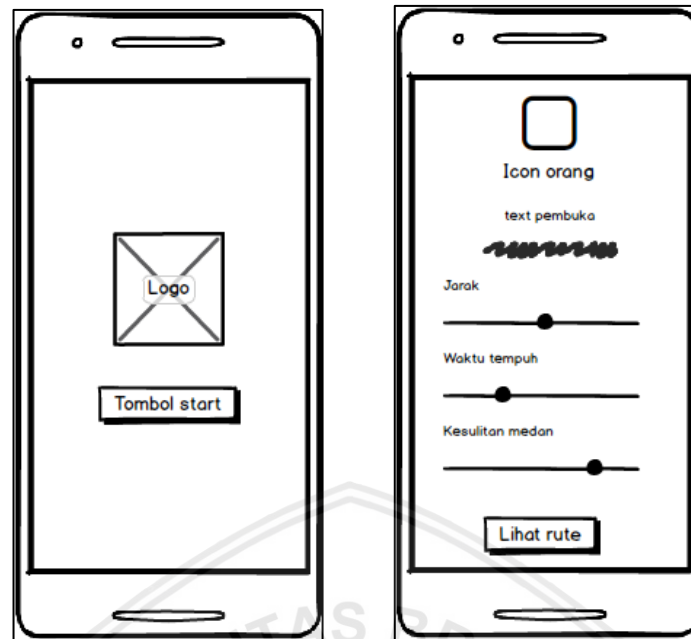
9	"namarute" : string,
10	"waktutempuh" : integer
11	}
12	}

4.2.5 Perancangan Antarmuka

Perancangan antarmuka ialah perancangan tampilan sistem rekomendasi rute gowes. Pada perancangan antarmuka ini, tampilan aplikasi akan terlihat jelas bagaimana aplikasi rekomendasi rute gowes dibangun nantinya. Perancangan antarmuka juga berguna untuk pengguna sistem karena dapat mengetahui alur dari sistem itu sendiri.

4.2.5.1 Perancangan Antarmuka Memberikan Bobot Nilai Untuk Setiap Kriteria

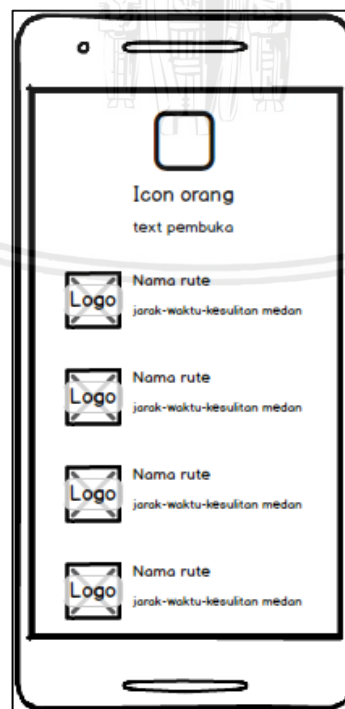
Perancangan antarmuka memberikan bobot nilai untuk setiap kriteria merupakan antarmuka pengguna yang berfungsi untuk memberikan bobot masing – masing kriteria untuk mendapatkan hasil rekomendasi rute yang sesuai. Pertama tama user diarahkan ke halaman pembuka dengan logo dan tombol mulai, selanjutnya pengguna menekan tombol tersebut dan diarahkan ke halaman kriteria. Pada halaman kriteria terdiri dari icon orang yaitu icon berupa gambar seseorang pria untuk membuat aplikasi terlihat interaktif dengan penggunaanya. Selain itu ada juga *text* pembuka dimana terdiri dari kalimat – kalimat pengguna yang menggambarkan bahwa aplikasi bisa bersifat interaktif. Dibawah *text* pembuka terdapat beberapa coretan yang menandakan ada kalimat – kalimat pendukung lainnya selain *text* pembuka diatas. Kemudian terdapat kriteria – kriteria yang akan digunakan untuk perhitungan TOPSIS. Pengguna memasukan bobot nilai dan menekan tombol lihat rute gowes. Pengisian bobot nilai untuk setiap kriteria ini dilakukan dengan menggeser *point* yang berbentuk bulat dan kemudian *point* tersebut akan menunjukan skala nilai 1 – 1 yang bisa pengguna pilih dengan sendirinya. Terakhir adalah tombol lihat rute yang akan menampilkan hasil dari rekomendasi rute gowes. Penggambaran secara jelas bisa di lihat pada Gambar 4.13 :



Gambar 4.13 Antarmuka Memberikan Bobot Nilai Untuk Setiap Kriteria

4.2.5.2 Perancangan Antarmuka Melihat Rekomendasi Rute Gowes

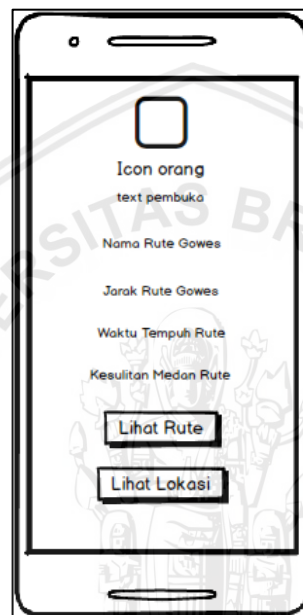
Perancangan antarmuka melihat rekomendasi rute gowes merupakan antarmuka pengguna yang berfungsi untuk menampilkan rute gowes yang paling sesuai dengan masukan pengguna pada halaman sebelumnya. Pada halaman ini, pengguna bisa melihat nama rute gowes, jarak rute gowes, waktu tempuh dan nilai kesulitan medan. Penggambaran secara jelas bisa di lihat pada Gambar 4.14 :



Gambar 4.14 Antarmuka Melihat Rekomendasi Rute Gowes

4.2.5.3 Perancangan Antarmuka Melihat Detail Rekomendasi Rute dan Petunjuk Arah

Perancangan antarmuka melihat detail rekomendasi rute dan petunjuk arah merupakan antarmuka pengguna yang berfungsi untuk melihat detail dari rekomendasi rute pada halaman sebelumnya dan untuk mengetahui arah rute gowes yang dimaksud. Pada halaman ini terdapat 2 tombol yang masing – masing memiliki fungsi berbeda yaitu tombol lihat rute gowes yang berfungsi untuk mengetahui titik awal dan titik akhir rute gowes. Sedangkan tombol lihat lokasi gowes berfungsi untuk mengetahui lokasi pengguna saat ini menuju titik awal rute gowes. Penggambaran secara jelas bisa di lihat pada Gambar 4.15 :



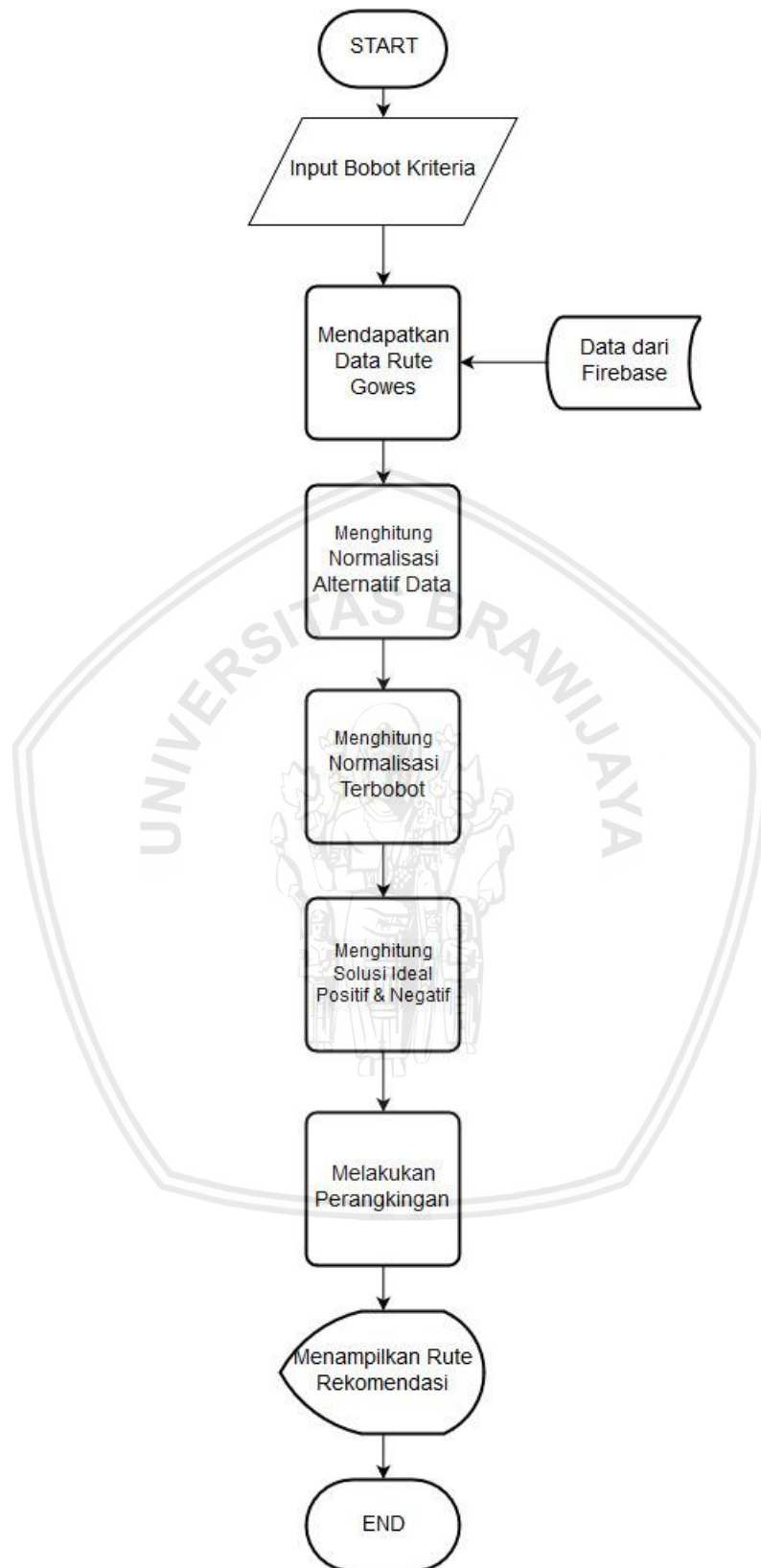
Gambar 4.15 Antarmuka Melihat Detail Rekomendasi Rute dan Petunjuk Arah

4.2.6 Perancangan Algoritme

Pada perancangan algoritme ini, penulis menjabarkan bagaimana aplikasi mengolah data dengan menerapkan metode TOPSIS yang tersusun atas perancangan TOPSIS dan kalkulasi manual.

4.2.6.1 Perancangan TOPSIS

Metode TOPSIS digunakan untuk melakukan perhitungan hasil rekomendasi rute gowes berlandaskan tiga kriteria utama yaitu jarak rute gowes, waktu tempuh dan kesulitan medan. Masing – masing kriteria tersebut akan diberikan bobot oleh pengguna. Setelah bobot telah diberikan selanjutnya akan dihitung normalisasi alternatif data, setelah alternatif data dilakukan normalisasi selanjutnya dilakukan normalisasi terbobot dan melakukan perhitungan untuk solusi ideal positif dan solusi ideal negatif. Hasil akhir yang ditampilkan merupakan rute rekomendasi yang telah melalui tahap perankingan. Untuk lebih jelas tentang tahap – tahap perhitungan TOPSIS mampu dilihat pada Gambar 4.16 :



Gambar 4.16 Flowchart TOPSIS

4.2.6.2 Perhitungan Manual

Pada subbab ini, penulis menjabarkan bagaimana kalkulasi manual dari metode TOPSIS yang digunakan pada penelitian. Sebelum masuk ke tahap pertama perhitungan manual metode TOPSIS, penulis akan menjelaskan kriteria apa yang diterapkan pada penelitian ini beserta keterangannya pada Tabel 4.9 :

Tabel 4.9 Penjelasan Kriteria Penelitian

Singkatan	Penjelasan
J	Jarak (Km)
WT	Waktu Tempuh (Menit)
KM	Kesulitan Medan (Skala 0 – 100)

Kriteria diatas akan digunakan oleh pengguna aplikasi untuk memberikan bobot nilai sehingga nantinya pengguna bisa memprioritaskan kriteria mana yang paling menonjol dalam memilih suatu rute gowes. Berikut merupakan contoh tahap – tahap perhitungan manual metode TOPSIS :

a. Menentukan alternatif data

Alternatif data disini merupakan beberapa rute gowes yang ada di kota Malang. Pada contoh perhitungan ini, penulis menggunakan data dari narasumber agar pembaca memahami alur dari metode TOPSIS ini. Data uji coba ini dapat di lihat pada Tabel 4.10 :

Tabel 4.10 Alternatif Data

No	Alternatif Data	J (KM)	WT (Menit)	KM (0-100)
1	Araya - Alun alun Batu	22	60	75
2	Rampal - Alun alun Malang	3	15	30
3	Araya - IXORA	5	20	10
4	Rampal - Bukit Tidar	12	45	80
5	Alun alun Malang - SMAN 10 Nala Taruna	10	30	70

b. Menghitung normalisasi alternatif data

Perhitungan normalisasi alternatif diterapkan pada semua kolom untuk setiap alternatif yang ada. Perhitungan lengkap terkait normalisasi alternatif ini tertera pada Tabel 4.11 :

Kolom J (KM) :

$$\sqrt{22^2 + 3^2 + 5^2 + 12^2 + 10^2} = 27.60434748$$

Kolom WT (Menit) :

$$\sqrt{60^2 + 15^2 + 20^2 + 45^2 + 30^2} = 84.55767263$$

Kolom KM (1-100) :

$$\sqrt{75^2 + 30^2 + 10^2 + 80^2 + 70^2} = 133.8842784$$

Contoh normalisasi untuk kolom J bari ke – 1 :

$$\frac{22}{27.60434748} = 0.796975912$$

Tabel 4.11 Normalisasi Alternatif

No	Alternatif Data	J (KM)	WT (Menit)	KM (0-100)
1	Araya - Alun alun Batu	0.796975912	0.709574875	0.560185265
2	Rampal - Alun alun Malang	0.108678533	0.177393719	0.224074106
3	Araya - IXORA	0.181130889	0.236524958	0.074691369
4	Rampal - Bukit Tidar	0.434714134	0.532181156	0.59753095
5	Alun alun Malang - SMAN 10 Nala Taruna	0.362261778	0.354787438	0.522839581

c. Menghitung normalisasi terbobot

Pada tahap ini, pengguna diminta untuk memasukan bobot nilai untuk tiga kriteria diatas. Pengguna akan memasukan bobot nilai untuk kriteria jarak, waktu tempuh serta kesulitan medan. Namun dalam contoh perhitungan ini penulis memasukan bobot nilai dengan data uji sembarang. Contoh bobot nilai untuk setiap kriteria mampu dilihat pada Tabel 4.12 :

Tabel 4.12 Bobot Nilai Kriteria

Kriteria	Bobot Nilai
Jarak (0-1)	0.5
Waktu Tempuh (0-1)	0.3
Kesulitan Medan (0-1)	0.2

Dari contoh bobot nilai diatas dapat dilihat bahwa pengguna aplikasi lebih memprioritaskan jarak tempuh dengan bobot nilai sebesar 0.5, kemudian disusul oleh kriteria kesulitan medan dengan bobot nilai sebesar 0.2 dan waktu tempuh dengan bobot nilai sebesar 0.3. Untuk melihat bobot nilai yang sudah di normalisasi dapat melihat pada Tabel 4.13 :

Contoh perhitungan normalisasi terbobot :

$$0.796975912 \times 0.5 = 0.398487956$$

Tabel 4.13 Normalisasi Terbobot

No	Alternatif Data	J (KM)	WT (Menit)	KM (0-100)
1	Kota Araya – Alun alun Batu	0.398487956	0.212872463	0.112037053
2	Rampal – Alun alun Batu	0.054339267	0.053218116	0.044814821
3	Kota Araya – Ixora	0.090565445	0.070957488	0.014938274
4	Rampal – Bukit Tidar	0.217357067	0.159654347	0.11950619
5	Idjen – Alun alun Malang	0.181130889	0.106436231	0.104567916

d. Menghitung solusi ideal positif dan negatif

Perhitungan untuk solusi ideal positif dan solusi ideal negatif ditentukan dengan melihat nilai maksimal dan nilai minimal untuk masing – masing kolom. Solusi ideal positif ditentukan dengan melihat nilai maksimal / nilai terbesar yang ada pada setiap kolom kriteria sedangkan solusi ideal negatif ditentukan dengan melihat nilai minimal / nilai terkecil yang ada pada masing – masing kolom kriteria. Solusi ideal positif dilambangkan dengan A+ dan solusi ideal negatif dilambangkan dengan A- . Untuk melihat hasil dari solusi ideal positif dan solusi ideal negatif mampu dilihat pada Tabel 4.14 :

Tabel 4.14 Solusi Ideal Positif dan Negatif

Solusi Ideal	J (KM)	WT (Menit)	KM (0-100)
A+	0.398487956	0.212872463	0.11950619
A-	0.054339267	0.053218116	0.014938274

e. Menghitung jarak alternatif dari solusi ideal positif dan negatif

Perhitungan untuk jarak alternatif untuk solusi ideal positif dan negatif adalah dengan cara mencari akar dari nilai solusi ideal positif dikurangi dengan normalisasi terbobot masing – masing kriteria dan dikurangi sebanyak jumlah kolom yang ada. Aturan yang sama juga berlaku untuk mencari jarak alternatif dari solusi ideal negatif. Untuk mempermudah pembacaan, jarak alternatif dari solusi ideal positif dilambangkan dengan SI+ sedangkan jarak alternatif dari solusi ideal negatif dilambangkan dengan SI- . Untuk melihat hasil perhitungan dari jarak alternatif solusi ideal positif dan negatif dapat dilihat pada Tabel 4.15:

Contoh perhitungan jarak alternatif dari solusi ideal positif (SI+) untuk baris ke – 1 :

$$\sqrt{(0.398487956 - 0.398487956)^2 + (0.212872463 - 0.212872463)^2 + (0.112037053 - 0.11950619)^2}$$

$$= 0.007469137$$

Contoh perhitungan jarak alternatif dari solusi ideal negatif (SI-) untuk baris ke - 1 :

$$\sqrt{(0.398487956 - 0.054339267)^2 + (0.212872463 - 0.053218116)^2 + (0.112037053 - 0.014938274)^2}$$

$$= 0.39160695$$

Tabel 4.15 Jarak Solusi Ideal Positif dan Negatif

SI+	SI-
0.007469	0.391607
0.386661	0.029877
0.354811	0.040336
0.188787	0.220993
0.242479	0.164139

f. Menghitung nilai preferensi

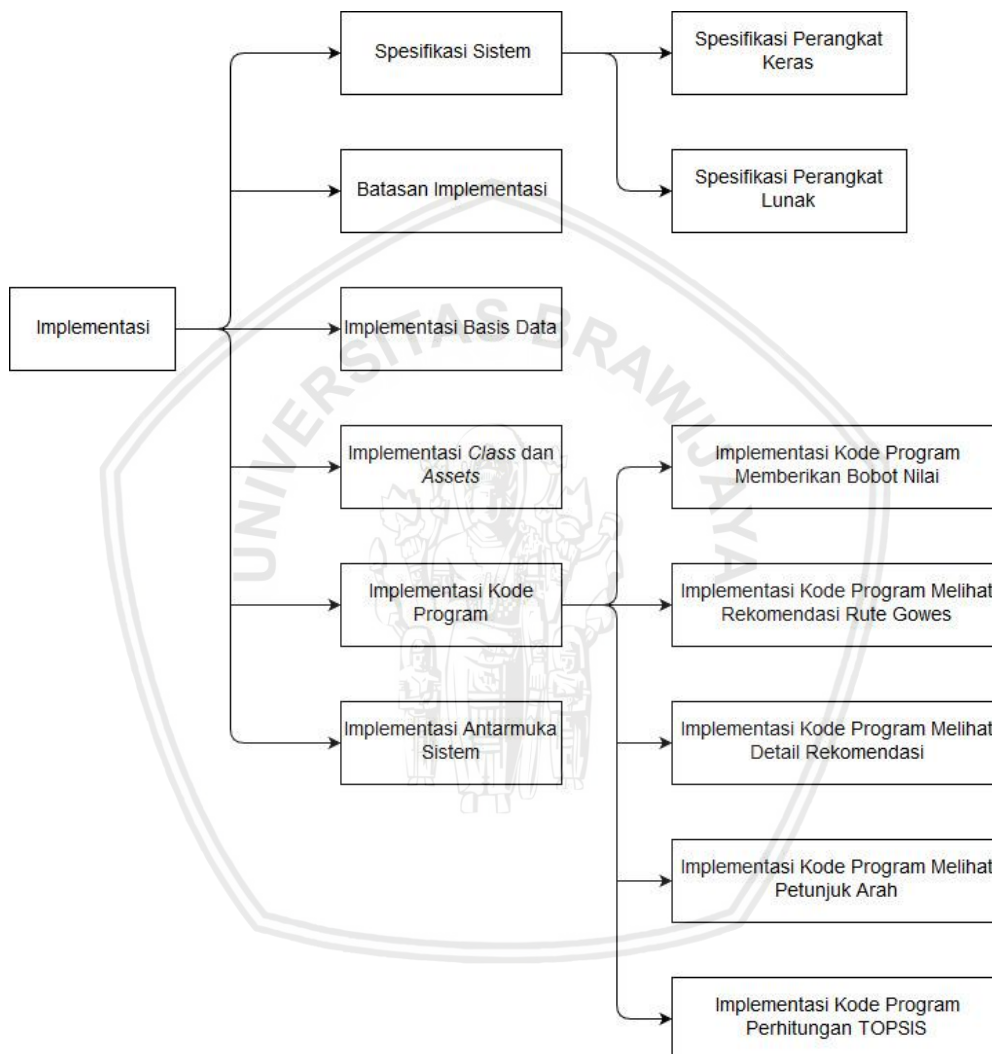
Perhitungan nilai preferensi diterapkan untuk melakukan perankingan dari rute yang sudah ada. Prosesnya adalah dengan membagi jumlah nilai SI- dengan jumlah dari SI+ dan SI- . Untuk melihat hasil dari preferensi, penulis sudah menjabarkannya dalam Tabel 4.16 :

Tabel 4.16 Nilai Preferensi

No	Alternatif Data	Nilai Preferensi
1	Araya - Alun alun Batu	0.981284
2	Rampal - Alun alun Malang	0.071726
3	Araya - IXORA	0.102079
4	Rampal - Bukit Tidar	0.539297
5	Alun alun Malang - SMAN 10 Nala Taruna	0.40367

BAB 5 IMPLEMENTASI

Bab implementasi menjabarkan mengenai implementasi dari sistem rekomendasi rute gowes yang akan dikembangkan. Bab ini berisikan spesifikasi sistem, batasan implementasi, implementasi *class* dan *assets*, implementasi kode program, dan implementasi antarmuka. Untuk melihat lebih jelas isi dari bab ini, penulis menjabarkannya pada Gambar 5.1 :



Gambar 5.1 Diagram Alir Implementasi

5.1 Spesifikasi Sistem

Spesifikasi sistem dibutuhkan dalam implementasi untuk menjadi panduan pengembangan sistem. Spesifikasi sistem terbagi menjadi dua yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak. Spesifikasi perangkat keras merupakan spesifikasi *hardware* yang dimanfaatkan dalam mengimplementasikan sistem rute gowes sedangkan spesifikasi perangkat lunak adalah *software* atau pun program lain yang mendukung dan digunakan dalam pengembangan sistem rekomendasi rute gowes.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang dimanfaatkan untuk pengembangan sistem rekomendasi rute gowes ini terbagi menjadi dua yakni spesifikasi perangkat keras komputer dan spesifikasi perangkat keras *android*. Spesifikasi perangkat keras komputer yang digunakan dapat dilihat pada Tabel 5.1 :

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Komponen	Spesifikasi
<i>Processor</i>	Intel Core i5-2410M CPU 2.30GHz
<i>Memory</i>	8192MB
<i>Display</i>	NVIDIA GeForce 410M

Perangkat keras *android* sendiri dimanfaatkan untuk membangun dan menjalankan sistem rekomendasi rute gowes. Spesifikasi perangkat keras *android* sendiri dapat dilihat pada Tabel 5.2 :

Tabel 5.2 Spesifikasi Perangkat Keras Android

Komponen	Spesifikasi
<i>Processor</i>	Qualcomm Snapdragon 636, 1.8GHz
<i>Memory</i>	3072 MB

5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang dimanfaatkan dalam pengembangan sistem rekomendasi rute gowes ini terbagi menjadi dua yaitu spesifikasi perangkat lunak komputer dan spesifikasi perangkat lunak *android*. Spesifikasi perangkat lunak komputer yang dimanfaatkan dapat dilihat pada Tabel 5.3 :

Tabel 5.3 Spesifikasi Perangkat Lunak Komputer

Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Alat Pengembangan	Android Studio 3.4
<i>Database</i>	<i>Google Firebase</i>
Bahasa Pemrograman	Java Android
Mesin Pencari	<i>Mozilla Firefox</i>

Perangkat lunak *android* sendiri digunakan untuk mengetahui sistem ini hendak dikembangkan pada sistem operasi versi berapa. Spesifikasi perangkat lunak *android* dapat dilihat pada Tabel 5.4 :

Tabel 5.4 Spesifikasi Perangkat Lunak *Android*

Komponen	Spesifikasi
<i>Platform</i>	Android OS 8.1.0

5.2 Batasan – batasan Implementasi

Dalam mengembangkan sistem rekomendasi rute gowes, penulis memiliki batasan – batasan dalam melakukan implementasi diantaranya adalah :

1. Sistem rekomendasi rute gowes diimplementasikan pada *platform android* dengan sistem operasi minimal 4.1 dan maksimal 9
2. Sistem dibuat dengan memakai alat pengembangan Android Studio dan memakai bahasa pemrograman Java.
3. Sistem menggunakan *Google Firebase* sebagai basis data

5.3 Implementasi Basis Data

Basis data yang dimanfaatkan penulis dalam penelitian ini adalah *Google Firebase* dengan memanfaatkan fitur *realtime database*. Adapun implementasi basis data pada *Firebase* menggunakan format JSON seperti Tabel 5.5 :

Tabel 5.5 Implementasi Basis Data

No	Source code
1	{
2	"data" : [null, {
3	"jarak" : 22,
4	"kesulitanmedan" : 75,
5	"lataakhir" : -7.8711591,
6	"lataawal" : -7.9365517,
7	"longakhir" : 112.5246605,
8	"longawal" : 112.6481222,
9	"nama" : "Araya - Alun alun Batu",
10	"waktutempuh" : 60
11	}, {
12	"jarak" : 3,
13	"kesulitanmedan" : 30,
14	"lataakhir" : -7.9825193,
15	"lataawal" : -7.9737781,
16	"longakhir" : 112.6286126,

17	"longawal" : 112.6381596,
18	"nama" : "Rampal - Alun alun Malang",
19	"waktutempuh" : 15
20	}, {
21	"jarak" : 5,
22	"kesulitanmedan" : 10,
23	"lataakhir" : -7.935279,
24	"lataawal" : -7.9365517,
25	"longakhir" : 112.6816848,
26	"longawal" : 112.6481222,
27	"nama" : "Araya - IXORA",
28	"waktutempuh" : 20
29	}, {
30	"jarak" : 12,
31	"kesulitanmedan" : 80,
32	"lataakhir" : -7.944205,
33	"lataawal" : -7.9737781,
34	"longakhir" : 112.582127,
35	"longawal" : 112.6381596,
36	"nama" : "Rampal - Bukit Tidar",
37	"waktutempuh" : 45
38	}, {
39	"jarak" : 10,
40	"kesulitanmedan" : 70,
41	"lataakhir" : -8.0385279,
42	"lataawal" : -7.9825193,
43	"longakhir" : 112.6524922,
45	"longawal" : 112.6286126,
46	"nama" : "Alun alun Malang - SMAN 10 Nala Taruna",
47	"waktutempuh" : 30
48	}, {
49	"jarak" : 23,
50	"kesulitanmedan" : 80,
51	"lataakhir" : -7.9967932,
52	"lataawal" : -7.9737781,
53	"longakhir" : 112.8024251,
54	"longawal" : 112.6381596,
55	"nama" : "Rampal - Coban Jidor",

```

56     "waktutempuh" : 60
57   }, {
58     "jarak" : 2,
59     "kesulitanmedan" : 20,
60     "lataakhir" : -7.9825193,
61     "lataawal" : -7.975548,
62     "longakhir" : 112.6286126,
63     "longawal" : 112.620709,
64     "nama" : "Idjen - Alun alun Malang",
65     "waktutempuh" : 10
66   }, {
67     "jarak" : 7,
68     "kesulitanmedan" : 25,
69     "lataakhir" : -7.9879801,
70     "lataawal" : -7.9365517,
71     "longakhir" : 112.6318118,
72     "longawal" : 112.6481222,
73     "nama" : "Araya - Pasar Besar",
74     "waktutempuh" : 25
75   }, {
76     "jarak" : 17,
77     "kesulitanmedan" : 20,
78     "lataakhir" : -8.0034972,
79     "lataawal" : -7.9365517,
80     "longakhir" : 112.7574147,
81     "longawal" : 112.6481222,
82     "nama" : "Araya - Tumpang",
83     "waktutempuh" : 45
84   } ]
85 }

```

Basis data yang penulis implementasikan terdiri dari data tabel, kesulitan medan, latitude akhir, latitude awal, longitude akhir, longitude akhir, nama serta waktu tempuh masing – masing rute yang ada di kota Malang. Latitude dan longitude merupakan koordinat dari *Google Maps* untuk mengetahui lokasi suatu tempat. Disini penulis menggunakan latitude longitude awal dan akhir untuk mendapatkan titik awal dan titik akhir lokasi sehingga nantinya rute akan menghubungkan dua titik tersebut. Selain itu penulis juga memberikan nama rute, waktu tempuh rute, kesulitan medan rute dan jarak masing – masing rute.

5.4 Implementasi *Class* dan *Assets*

Implementasi *class* dan *assets* berisikan berkas - berkas yang membentuk sistem rekomendasi rute gowes. Berkas - berkas ini sendiri memiliki dua *format* yaitu java dan xml. Untuk berkas *berformat* java digunakan untuk menjalankan kode program dari sistem sedangkan berkas *berformat* xml digunakan untuk membuat tampilan antarmuka sistem. Penulis sudah menjabarkan berkas – berkas tersebut pada Tabel 5.6 :

Tabel 5.6 Implementasi *Class* dan *Assets*

No	Nama <i>Package</i>	Nama <i>Class</i>	Nama <i>Layout</i>
1	com.example. sistemrekomendasigow es	MainActivity.java	activity_main.xml
2	com.example. sistemrekomendasigow es	Kriteria.java	activity_kriteria.xml
3	com.example. sistemrekomendasigow es	Daftar_Rute.java	activity_daftar_rute.xml
4	com.example. sistemrekomendasigow es	Detail_Rute.java	activity_detail_rute.xml
5	com.example. sistemrekomendasigow es	Rincian_Rute.java	-
6	com.example. sistemrekomendasigow es	Adapter_Daftar_Rute.java	-

5.5 Implementasi Kode Program

Implementasi kode program ialah inti dari pengembangan sistem rekomendasi rute gowes sehingga keluaran yang dihasilkan bisa berjalan dengan baik pada *platform android*. Implementasi kode program terbagi menjadi beberapa bagian yakni implementasi kode program memberikan bobot nilai, implementasi kode program melihat rekomendasi rute gowes, implementasi kode program melihat detail rekomendasi, implementasi kode program melihat petunjuk arah, serta implementasi kode program perhitungan TOPSIS.

5.5.1 Implementasi Kode Program Memberikan Bobot Nilai

Fungsionalitas memberikan bobot nilai dapat dipanggil saat pengguna telah menekan tombol Mulai Gowes pada MainActivity.java. Adapun kode program untuk fungsionalitas ini terdapat pada *class* Kriteria.java dan mampu dilihat pada Tabel 5.7 :

Tabel 5.7 Implementasi Kode Program Memberikan Bobot Nilai

No	Kriteria.java
1	<code>kriteria_jarak.setOnProgressChangeListener(new</code>
2	<code>DiscreteSeekBar.OnProgressChangeListener() {</code>
3	<code> @Override</code>
4	<code> public void onProgressChanged(DiscreteSeekBar</code>
5	<code> seekBar, int value, boolean fromUser) {</code>
6	<code> progresNilaiJarak = ((float)value / 10.0);</code>
7	<code> }</code>
8	<code> @Override</code>
9	<code> public void onStartTrackingTouch(DiscreteSeekBar</code>
10	<code> seekBar) {</code>
11	<code> }</code>
12	<code> @Override</code>
13	<code> public void onStopTrackingTouch(DiscreteSeekBar</code>
14	<code> seekBar) {</code>
15	<code> Toast toast =</code>
16	<code> Toast.makeText(getBaseContext(), "Bobot Jarak = " +</code>
17	<code> progresNilaiJarak, Toast.LENGTH_SHORT);</code>
18	<code> toast.setGravity(Gravity.CENTER </code>
19	<code> Gravity.CENTER_HORIZONTAL, 0, 0);</code>
20	<code> toast.show();</code>
21	<code> }</code>
22	<code> });</code>
23	<code>kriteria_kesulitan_medan.setOnProgressChangeListener(new</code>
24	<code>DiscreteSeekBar.OnProgressChangeListener() {</code>
25	<code> @Override</code>
26	<code> public void onProgressChanged(DiscreteSeekBar</code>
27	<code> seekBar, int value, boolean fromUser) {</code>
28	<code> progresNilaiKesulitan = ((float)value /</code>
29	<code> 10.0);</code>
30	<code> }</code>
31	<code> @Override</code>
32	<code> public void onStartTrackingTouch(DiscreteSeekBar</code>
33	<code> seekBar) {</code>
34	<code> }</code>
35	<code> @Override</code>
	<code> public void onStopTrackingTouch(DiscreteSeekBar</code>
	<code> seekBar) {</code>
	<code> Toast toast =</code>
	<code> Toast.makeText(getBaseContext(), "Bobot Kesulitan Medan = "</code>
	<code> + progresNilaiKesulitan, Toast.LENGTH_SHORT);</code>

```

36         toast.setGravity(Gravity.CENTER |
Gravity.CENTER_HORIZONTAL, 0, 0);
37         toast.show();
39     }
39     });
40
41
42     kriteria_waktu_tempuh.setOnProgressChangeListener(new
43     DiscreteSeekBar.OnProgressChangeListener() {
44         @Override
45         public void onProgressChanged(DiscreteSeekBar
seekBar, int value, boolean fromUser) {
46             progresNilaiWaktu = ((float)value / 10.0);
47         }
48
49         @Override
50         public void onStartTrackingTouch(DiscreteSeekBar
seekBar) {
51
52         }
53
54         @Override
55         public void onStopTrackingTouch(DiscreteSeekBar
seekBar) {
56             Toast toast =
Toast.makeText(getBaseContext(), "Bobot Waktu Tempuh = " +
progresNilaiWaktu, Toast.LENGTH_SHORT);
57             toast.setGravity(Gravity.CENTER |
Gravity.CENTER_HORIZONTAL, 0, 0);
58             toast.show();
59         }
60     });

```

Berikut adalah penjelasan dari kode program Tabel 5.7 :

1. Baris 1 – 2

Merupakan method untuk memperoleh nilai dari kriteria jarak dengan masukan dari pengguna dengan cara melakukan *slide* pada *seekbar*

2. Baris 3 – 6

Merupakan method untuk memasukan nilai jarak ke dalam *variable global* *progresNilaiJarak*

3. Baris 13 – 19

Merupakan method untuk menampilkan nilai yang telah dimasukan pengguna setelah pengguna melepas *slide* kriteria jarak

4. Baris 22

Merupakan method untuk mendapatkan nilai dari kriteria kesulitan medan dengan masukan dari pengguna dengan cara melakukan *slide* pada *seekbar*

5. Baris 23 – 26

Merupakan method untuk memasukan nilai kesulitan medan ke dalam *variable global* *progresNilaiKesulitan*

6. Baris 33 – 39

Merupakan method untuk menampilkan nilai yang telah dimasukan pengguna setelah pengguna melepas *slide* kriteria kesulitan medan

7. Baris 42 – 43

Merupakan method untuk mendapatkan nilai dari kriteria waktu tempuh dengan masukan dari pengguna dengan cara melakukan *slide* pada *seekbar*

8. Baris 44 – 47

Merupakan method untuk memasukan nilai jarak ke dalam *variable global* *progresNilaiWaktu*

9. Baris 54 – 60

Merupakan method untuk menampilkan nilai yang telah dimasukan pengguna setelah pengguna melepas *slide* kriteria waktu tempuh

Setelah nilai bobot telah dimasukan pengguna maka ketika pengguna menekan tombol Lihat Rute Gowes akan menjalankan *method setOnClickListener* dan akan mengirimkan nilai tersebut ke *class Daftar_Rute*. *Method setOnClickListener* dapat dilihat pada Tabel 5.8 :

Tabel 5.8 Method *setOnClickListener* pada Class *Daftar_Rute*

No	Method <i>setOnClickListener</i>
1	<code>btn_lihat_rute.setOnClickListener(new</code>
2	<code>View.OnClickListener() {</code>
3	<code>@Override</code>
4	<code>public void onClick(View v) {</code>
5	<code>Intent intent = new Intent(Kriteria.this,</code>
6	<code>Daftar_Rute.class);</code>
7	<code>Double bobotJarak = new</code>
8	<code>Double(progresNilaiJarak);</code>
9	<code>Double bobotWaktu = new</code>
10	<code>Double(progresNilaiWaktu);</code>
11	<code>Double bobotKesulitan = new</code>
12	<code>Double(progresNilaiKesulitan);</code>
13	<code>Bundle bundle = new Bundle();</code>
14	<code>bundle.putDouble("bobot_jarak",</code>
15	<code>bobotJarak);</code>
16	<code>bundle.putDouble("bobot_waktu",</code>
17	<code>bobotWaktu);</code>
18	<code>bundle.putDouble("bobot_kesulitan",</code>
19	<code>bobotKesulitan);</code>
20	<code>intent.putExtras(bundle);</code>
21	<code>startActivity(intent);</code>
22	<code>}</code>
23	<code>});</code>

Berikut adalah penjelasan dari kode program Tabel 5.8 :

1. Baris 1

Merupakan *method* yang digunakan untuk memberikan aksi ketika tombol Lihat Rute Gowes ditekan

2. Baris 4

Merupakan deklarasi perpindahan dari *class* Kriteria ke *class* Daftar_Rute dengan menggunakan *library Android Studio*

3. Baris 5 – 7

Merupakan proses mengubah tipe data dari masukan pengguna untuk kriteria jarak, waktu tempuh dan kesulitan medan menjadi bentuk tipe data *double*

4. Baris 8 – 12

Merupakan proses memasukan data nilai masing – masing bobot ke dalam *library bundle* agar dapat di baca pada *class* Daftar_Rute

5. Baris 13

Merupakan kode untuk memulai perpindahan dari *class* Kriteria ke *class* Daftar_Rute

5.5.2 Implementasi Kode Program Melihat Rekomendasi Rute Gowes

Setelah *class* Kriteria menjalankan *method onClickListener* maka aplikasi akan berpindah ke *class* Daftar_Rute untuk menampilkan hasil dari perhitungan TOPSIS kepada pengguna. Adapun kode program untuk fungsionalitas ini bisa dilihat pada Tabel 5.9 :

Tabel 5.9 Implementasi Kode Program Melihat Rekomendasi Rute

No	Daftar_Rute.java
1	Bundle bundle = getIntent().getExtras();
2	
3	if(bundle!=null){
4	bobotJarak = bundle.getDouble("bobot_jarak");
5	bobotWaktu = bundle.getDouble("bobot_waktu");
6	bobotKesulitan = bundle.getDouble("bobot_kesulitan");
7	} else {
8	Log.d("gagal", "data tidak sampai ");
9	}
10	recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
11	RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(Daftar_Rute.this);
12	recyclerView.setLayoutManager(layoutManager);
13	adapter = new Adapter_Daftar_Rute((ArrayList<List<String>>) arrayDataRute, this);
14	
15	recyclerView.setAdapter(adapter);

16	progress=new ProgressDialog(this);
17	progress.setTitle("Proses Perhitungan");
18	progress.setMessage("Mengolah data ...");
19	
20	progress.setProgressStyle(ProgressDialog.STYLE_SPINNER);
21	progress.show();
22	Runnable progressRunnable = new Runnable() {
23	
24	@Override
25	public void run() {
26	progress.cancel();
27	}
28	};
29	Handler pdCanceller = new Handler();
30	pdCanceller.postDelayed(progressRunnable, 2000);

Berikut adalah penjelasan dari kode program Tabel 5.9 :

1. Baris 1 – 9

Merupakan proses menerima data bobot nilai yang telah di masukan pengguna di class Kriteria sebelumnya

2. Baris 10 – 15

Merupakan proses menampilkan hasil perhitungan TOPSIS ke tampilan Daftar_Rute dengan menggunakan library *recyclerview*

3. Baris 16 – 30

Merupakan proses menampilkan *dialog loading* pada tampilan Daftar_Rute dengan lama waktu 2 detik

Untuk mengatur tampilan pada *recyclerview* diperlukan *class* Adapter_Daftar_Rute. *Class* ini digunakan untuk mengatur tampilan dari *recyclerview* dan mengatur data yang akan ditampilkan. Adapun *method* yang digunakan pada *class* Adapter_Daftar_Rute adalah *method onBindViewHolder* yang bisa dilihat pada Tabel 5.10 :

Tabel 5.10 Method onBindViewHolder

No	Method onBindViewHolder
1	public void onBindViewHolder(Rincian_RuteViewHolder holder,
	final int position) {
2	
3	holder.namaRute.setText(dataList.get(position).get(1));
4	holder.jarakRute.setText(dataList.get(position).get(2) +
	" km");
5	
6	holder.kesulitan_medanRute.setText(dataList.get(position).get(3)+
	"/100");
7	
8	holder.waktu_tempuhRute.setText(dataList.get(position).get(4) + "
	menit");
9	
10	holder.parentLayout.setOnClickListener(new
	View.OnClickListener() {

11	@Override
12	public void onClick(View v) {
13	Intent intent = new Intent(context,
14	Detail_Rute.class);
15	intent.putExtra("jarak_rute",
16	dataList.get(position).get(2));
17	intent.putExtra("waktu_tempuh_rute",
18	dataList.get(position).get(4));
19	intent.putExtra("kesulitan_medan_rute",
20	dataList.get(position).get(3));
21	intent.putExtra("nama_rute",
22	dataList.get(position).get(1));
23	intent.putExtra("lat_awal",
24	dataList.get(position).get(6));
25	intent.putExtra("long_akhir",
	dataList.get(position).get(8));
	intent.putExtra("long_akhir",
	dataList.get(position).get(9));
	context.startActivity(intent);
	}
	});
	}

Berikut adalah penjelasan dari kode program Tabel 5.10 :

1. Baris 1
Merupakan deklarasi *method onBindViewHolder*
2. Baris 3 – 8
Merupakan proses memasukan data – data yang terdapat dalam *Firebase* kedalam *recyclerview* yang berada di dalam *class* Daftar_Rute
3. Baris 10 – 13
Merupakan proses mengirimkan data terpilih ke *class* Detail_Rute
4. Baris 14 – 21
Merupakan proses mengirimkan data nama rute, jarak, kesulitan medan, waktu tempuh, latitude dan longitude awal serta latitude dan longitude akhir ke *class* Detail_Rute dengan menggunakan kata kunci tertentu
5. Baris 22
Merupakan proses memulai mengirimkan data – data sebelumnya

5.5.3 Implementasi Kode Program Melihat Detail Rekomendasi

Setelah *class* Adapter_Daftar_Rute mengirimkan data – data terkait rute gowes maka selanjutnya pada *class* Detail_Rute akan menampilkan data – data tersebut. Proses menampilkan data – data tersebut terdapat pada Tabel 5.11 :

Tabel 5.11 Implementasi Kode Program Melihat Detail Rute

No	Detail_Rute.java
1	private void getIncomingIntent() {
2	
3	if (getIntent().hasExtra("nama_rute") &&
4	getIntent().hasExtra("jarak_rute") &&
5	getIntent().hasExtra("waktu_tempuh_rute") &&
6	&& getIntent().hasExtra("kesulitan_medan_rute")
7	&& getIntent().hasExtra("lat_awal") &&
8	getIntent().hasExtra("lat_akhir") &&
9	getIntent().hasExtra("long_akhir")) {
10	
11	String namaRute =
12	getIntent().getStringExtra("nama_rute");
13	String kesulitanmedanRute =
14	getIntent().getStringExtra("kesulitan_medan_rute");
15	String waktutempuhRute =
16	getIntent().getStringExtra("waktu_tempuh_rute");
17	String jarakRute =
18	getIntent().getStringExtra("jarak_rute");
19	String latAwal =
20	getIntent().getStringExtra("lat_awal");
21	String longAwal =
22	getIntent().getStringExtra("long_awal");
23	String latAkhir =
24	getIntent().getStringExtra("lat_akhir");
25	String longAkhir=
26	getIntent().getStringExtra("long_akhir");
27	
28	setData(namaRute, jarakRute, waktutempuhRute,
29	kesulitanmedanRute,latAwal,longAwal,latAkhir,longAkhir);
30	}
31	}
32	
33	private void setData(String namaRute, String jarakRute,
34	String waktutempuhRute, String kesulitanmedanRute ,
35	String latAwal, String longAwal,
36	String latAkhir, String longAkhir) {
37	
38	TextView namarute =
39	findViewById(R.id.nama_rute_text);
40	namarute.setText(namaRute);
41	
42	TextView jarakrute =
43	findViewById(R.id.jarak_rute_text);
44	jarakrute.setText("Jarak Rute : " + jarakRute + " KM");
45	
46	TextView waktutempuh =
47	findViewById(R.id.waktu_tempuh_rute_text);
48	waktutempuh.setText("Ditempuh dalam waktu : " +
49	waktutempuhRute + " menit");
50	
51	TextView kesulitanmedan =
52	findViewById(R.id.kesulitan_medan_rute_text);
53	kesulitanmedan.setText("Dengan kesulitan medan " +
54	kesulitanmedanRute + "/100");

41	LatAwalDouble = Double.valueOf(latAwal);
	LongAwalDouble = Double.valueOf(longAwal);
42	LatAkhirDouble = Double.valueOf(latAkhir);
	LongAkhirDouble = Double.valueOf(longAkhir);
43	
44	}
45	
46	
47	
48	
49	

Berikut adalah penjelasan dari kode program Tabel 5.11 :

1. Baris 1
Merupakan *method* untuk menerima semua data – data rute gowes
2. Baris 3 – 11
Merupakan proses seleksi untuk apakah kata kunci dalam *class* Detail_Rute sama dengan yang dikirimkan dari *class* Adapter_Daftar_Rute
3. Baris 13 – 20
Merupakan proses memasukan data yang diterima kedalam *variable* lokal dengan tipe data *string*
4. Baris 22
Merupakan proses memanggil *method* *setData* dengan parameter *variable* lokal yang sebelumnya telah dideklarasikan
5. Baris 26 – 30
Merupakan *method* untuk menampilkan data – data rute gowes ke dalam tampilan *class* Detail_Rute
6. Baris 32 – 42
Merupakan proses untuk menampilkan data rute gowes ke dalam *text* yang berada pada *class* Detail_Rute
7. Baris 44 – 47
Merupakan proses untuk merubah tipe data latitude dan longitude yang semula bertipe data *string* menjadi bertipe data *double*

5.5.4 Implementasi Kode Program Melihat Petunjuk Arah

Fungsionalitas melihat petunjuk arah berada dalam *class* Detail_Rute namun untuk mendapatkan petunjuk arah harus menekan dua tombol di dalam tampilan layar. Setelah tombol ditekan maka secara otomatis pengguna akan membuka *Google Maps* yang terdapat pada perangkat android .Saat tombol ditekan maka akan menjalankan *method* *onClickListener* yang bisa dilihat pada Tabel 5.12 :

Tabel 5.12 Method *onClickListener* pada Class *Detail_Rute*

No	Method <i>onClickListener</i>
1	FusedLocationProviderClient mFusedLocation =
2	LocationServices.getFusedLocationProviderClient(this);
3	if (ActivityCompat.checkSelfPermission(this,
4	Manifest.permission.ACCESS_FINE_LOCATION) !=
5	PackageManager.PERMISSION_GRANTED &&
6	ActivityCompat.checkSelfPermission(this,
7	Manifest.permission.ACCESS_COARSE_LOCATION) !=
8	PackageManager.PERMISSION_GRANTED) {
9	
10	return;
11	}
12	mFusedLocation.getLastLocation().addOnSuccessListener(this,
13	new OnSuccessListener<Location>() {
14	@Override
15	public void onSuccess(Location location) {
16	if (location != null) {
17	latitudeUser = location.getLatitude();
18	longitudeUser = location.getLongitude();
19	Log.d("My Current location", "Lat : " +
20	latitudeUser + " Long : " + longitudeUser);
21	}
22	}
23	});
24	btn_lihat_lokasi = (Button)
25	findViewById(R.id.btn_lihat_lokasi);
26	btn_lihat_rute = (Button)
27	findViewById(R.id.btn_lihat_rute);
28	btn_lihat_lokasi.setOnClickListener(new
29	View.OnClickListener() {
30	@Override
31	public void onClick(View v) {
32	Intent intent = new
33	Intent(android.content.Intent.ACTION_VIEW,
34	Uri.parse("http://maps.google.com/maps?saddr="+latitudeUser
35	+", "+longitudeUser+"&daddr="+LatAwalDouble+", "+LongAwalDoub
36	le));
37	startActivity(intent);
38	}
39	});
40	btn_lihat_rute.setOnClickListener(new
41	View.OnClickListener() {
42	@Override
43	public void onClick(View v) {
44	Intent intent = new
45	Intent(android.content.Intent.ACTION_VIEW,
46	Uri.parse("http://maps.google.com/maps?saddr="+LatAwalDoub
47	le+", "+LongAwalDouble+"&daddr="+LatAkhirDouble+", "+LongAkhir
48	Double));
49	startActivity(intent);
50	}

39	});
40	

Berikut adalah penjelasan dari kode program Tabel 5.12 :

1. Baris 1

Merupakan deklarasi fungsi dalam *android studio* untuk mendapatkan lokasi pengguna

2. Baris 2 – 16

Merupakan proses untuk mendapatkan data latitude dan longitude pengguna yang nantinya akan di simpan dalam *variable global* latitudeUser dan longitudeUser sehingga nantinya dalam *Google Maps* dapat mengetahui posisi pengguna

3. Baris 18 – 19

Merupakan proses deklarasi tombol dalam *android studio*

4. Baris 21 – 29

Merupakan proses untuk membuka *Google Maps* dengan mengirimkan *link* berisi latitude dan longitude user serta latitude dan longitude awal rute gowes yang di pilih pada tombol lihat lokasi

5. Baris 31 – 40

Merupakan proses untuk membuka *Google Maps* dengan mengirimkan *link* berisi latitude dan longitude titik awal rute serta latitude dan longitude titik akhir rute gowes yang di pilih pada tombol lihat rute

5.5.5 Implementasi Kode Program Perhitungan TOPSIS

Sebelum melakukan proses perhitungan TOPSIS , terlebih dahulu dilakukan pengambilan data rute – rute gowes yang terdapat di dalam *Firebase* dan dilakukan inisialisasi maupun deklarasi array yang akan digunakan nantinya dalam perhitungan TOPSIS. Ada pun kode program untuk mengambil data rute – rute gowes tersebut terdapat pada *class* Daftar_Rute pada Tabel 5.13 :

Tabel 5.13 Inisialisasi *Firebase* dan ArrayList

No	Daftar_Rute.java
1	//inisialisasi
2	data_rute = new Rincian_Rute();
3	
4	//array 2 dimensi data rute
5	arrayDataRute = new ArrayList<List<String>>();
6	arrayNormalisasi = new ArrayList<List<Double>>();
7	arrayTerbobot = new ArrayList<List<Double>>();
8	arraySolusiIdeal = new ArrayList<List<Double>>();
9	arrayJarakIdeal = new ArrayList<List<Double>>();
10	arrayNilaiPrefensi = new ArrayList<List<Double>>();
11	
12	// Write a message to the database

13	database = FirebaseDatabase.getInstance();
14	reference = database.getReference("data");

Berikut adalah penjelasan dari kode program Tabel 5.13 :

1. Baris 1 – 2

Merupakan inisialisasi *arraylist* yang menggunakan pemodelan dari *class* Rincian_Rute

2. Baris 4 – 10

Merupakan inisialisasi *arraylist* dua dimensi yang akan digunakan untuk perhitungan TOPSIS dan merupakan *variable global*

3. Baris 12 – 14

Merupakan inisialisasi *Firebase* dalam *android studio* untuk bisa mengakses *realtime database*. Dalam kode program ini, penulis mengakses basis data dengan nama "data"

Selanjutnya adalah proses untuk memperoleh data dari *Firebase* dan akan disimpan dalam *arraylist* dua dimensi sehingga nantinya akan mempermudah proses perhitungan TOPSIS itu sendiri. Proses pengambilan data tersebut terdapat pada *method getDataRute* dan dapat dilihat pada Tabel 5.14 :

Tabel 5.14 Method *getDataRute*

No	Method <i>getDataRute</i>
1	private void <i>getDataRute</i> () {
2	reference.addValueEventListener(new ValueEventListener() {
3	@Override
4	public void onDataChange(@NonNull DataSnapshot
5	dataSnapshot) {
6	for (DataSnapshot ds : dataSnapshot.getChildren()) {
7	Map<String, Object> map = (Map<String, Object>)
8	ds.getValue();
9	data_rute.setId(ds.getKey());
10	data_rute.setNama_rute(map.get("nama").toString());
11	data_rute.setJarak(map.get("jarak").toString());
12	data_rute.setKesulitan_medan(map.get("kesulitanmedan").toString());
13	data_rute.setWaktu_tempuh(map.get("waktutempuh").toString());
14	data_rute.setLatawal(map.get("latawal").toString());
15	data_rute.setLongawal(map.get("longawal").toString());
16	data_rute.setLataakhir(map.get("lataakhir").toString());
17	data_rute.setLongakhir(map.get("longakhir").toString());
18	kolomPrefensi = "0";
19	arrayDataRute.add(Arrays.asList(data_rute.getId(),
20	
21	
22	
23	
24	
25	
26	

```

27         data_rute.getJarak(), data_rute.getNama_rute(),
28         data_rute.getWaktu_tempuh(), data_rute.getKesulitan_medan(),
29         kolomPrefensi, data_rute.getLatawal(), data_rute.getLongawal(),
30         data_rute.getLongakhir(), data_rute.getLongakhir()));
31     }
32 }
33
34     Log.d("rute", "Data Rute : " + arrayDataRute);
35     TOPSIS(arrayDataRute);
36 }
37
38     @Override
39     public void onCancelled(@NonNull DatabaseError
40     databaseError) {
41     }
42 }
43 }

```

Berikut ini adalah penjelasan dari kode program Tabel 5.14 :

1. Baris 1 – 2

Merupakan inisialisasi *method getDataRute*

2. Baris 5 – 6

Merupakan proses perulangan dalam membaca data untuk masing – masing turunan dari basis data “data” dan akan disimpan dalam bentuk objek bertipe data *string*

3. Baris 7 – 24

Merupakan proses pemodelan data – data yang ada didalam *Firestore* kedalam *class Rincian_Rute*

4. Baris 26 – 30

Merupakan proses pengisian *arraylist* dua dimensi dengan nama *arrayDataRute* yang berisi data – data rute gowes yang sudah dimodelkan sebelumnya

5. Baris 34 – 35

Merupakan proses pemanggilan *method TOPSIS* dengan parameter *arraylist* dua dimensi *arrayDataRute*

Setelah data dari *Firestore* dimasukan ke dalam *arraylist* dua dimensi dengan nama *arrayDataRute* maka selanjutnya adalah proses perhitungan *TOPSIS* itu sendiri. *Method TOPSIS* berfungsi sebagai tempat perhitungan algoritme *TOPSIS* dan adapun kode program untuk *method TOPSIS* bisa dilihat pada Tabel 5.15 :

Tabel 5.15 Method TOPSIS

No	Method TOPSIS
1	private void TOPSIS(List<List<String>> arrayDataRute){
2	
3	//untuk proses pembagi
4	pembagiJarak =
	Math.sqrt(Math.pow(Double.valueOf(arrayDataRute.get(0).get(2)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(1).get(2)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(2).get(2)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(3).get(2)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(4).get(2)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(5).get(2)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(6).get(2)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(7).get(2)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(8).get(2)),2));
5	pembagiKesulitanMedan =
	Math.sqrt(Math.pow(Double.valueOf(arrayDataRute.get(0).get(3)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(1).get(3)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(2).get(3)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(3).get(3)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(4).get(3)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(5).get(3)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(6).get(3)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(7).get(3)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(8).get(3)),2));
6	pembagiWaktuTempuh =
	Math.sqrt(Math.pow(Double.valueOf(arrayDataRute.get(0).get(4)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(1).get(4)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(2).get(4)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(3).get(4)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(4).get(4)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(5).get(4)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(6).get(4)),2)
	+
	Math.pow(Double.valueOf(arrayDataRute.get(7).get(4)),2)+
	Math.pow(Double.valueOf(arrayDataRute.get(8).get(4)),2));
7	//untuk normalisasi
8	for (int a = 0; a < arrayDataRute.size(); a++){
9	normalisasiJarak =
	Double.valueOf(arrayDataRute.get(a).get(2))/pembagiJarak;
10	normalisasiKesulitan
	Double.valueOf(arrayDataRute.get(a).get(3))/pembagiKesulitanMedan;

```

11         normalisasiWaktu =
Double.valueOf(arrayDataRute.get(a).get(4))/pembagiWaktuTempu
h;

12     arrayNormalisasi.add(Arrays.asList(normalisasiJarak,normalisa
siKesulitan,normalisasiWaktu));
13     }
14
15     Log.d("normalisasi", "Normalisasi : " +
arrayNormalisasi.get(0).get(0));
16
17     //untuk nilai terbobot
18     for (int a = 0; a < arrayNormalisasi.size(); a++){
19         terbobotJarak =
Double.valueOf(arrayNormalisasi.get(a).get(0))*bobotJarak;
20         terbobotKesulitan =
Double.valueOf(arrayNormalisasi.get(a).get(1))*bobotKesulitan
;
21         terbobotWaktu =
Double.valueOf(arrayNormalisasi.get(a).get(2))*bobotWaktu;
22     arrayTerbobot.add(Arrays.asList(terbobotJarak,terbobotKesulit
an,terbobotWaktu));
23
24     }
25     Log.d("normalisasi", "Terbobot : " +
arrayTerbobot.get(0));
26
27     //untuk solusi ideal positif dan negatif
28
29     maxValueJarak = arrayTerbobot.get(0).get(0);
30     minValueJarak = arrayTerbobot.get(0).get(0);
31     maxValueKesulitan = arrayTerbobot.get(0).get(1);
32     minValueKesulitan = arrayTerbobot.get(0).get(1);
33     maxValueWaktu = arrayTerbobot.get(0).get(2);
34     minValueWaktu = arrayTerbobot.get(0).get(2);
35
36     //untuk solusi ideal positif negatif jarak
37     for (int a=1; a < arrayTerbobot.size(); a++){
38         if (arrayTerbobot.get(a).get(0) > maxValueJarak){
39             maxValueJarak = arrayTerbobot.get(a).get(0);
40         } if (arrayTerbobot.get(a).get(0) <
minValueJarak){
41             minValueJarak = arrayTerbobot.get(a).get(0);
42         }
43     }
44
45     //untuk solusi ideal positif negatif kesulitan medan
46     for (int a=1; a < arrayTerbobot.size(); a++){
47         if (arrayTerbobot.get(a).get(1) >
maxValueKesulitan){
48             maxValueKesulitan =
arrayTerbobot.get(a).get(1);
49         } if (arrayTerbobot.get(a).get(1) <
minValueKesulitan){
50             minValueKesulitan =
arrayTerbobot.get(a).get(1);
51         }
52     }

```

```

53
54         //untuk solusi ideal positif negatif waktu tempuh
55         for (int a=1; a < arrayTerbobot.size(); a++){
56             if (arrayTerbobot.get(a).get(2) > maxValueWaktu){
57                 maxValueWaktu = arrayTerbobot.get(a).get(2);
58             } if (arrayTerbobot.get(a).get(2) <
minValueWaktu){
59                 minValueWaktu = arrayTerbobot.get(a).get(2);
60             }
61         }
62
63
64         arraySolusiIdeal.add(Arrays.asList(maxValueJarak,maxValueKesu
litan,maxValueWaktu));
65
66         arraySolusiIdeal.add(Arrays.asList(minValueJarak,minValueKesu
litan,minValueWaktu));
67         Log.d("solusi ideal", "array solusi ideal : " +
arraySolusiIdeal);
68
69         //untuk jarak ideal positif dan negatif
70         for (int a=0; a < arrayTerbobot.size(); a++){
71             jarakIdealPositif =
Math.sqrt(Math.pow(arrayTerbobot.get(a).get(0)-
arraySolusiIdeal.get(0).get(0),2)
+Math.pow(arrayTerbobot.get(a).get(1)-
arraySolusiIdeal.get(0).get(1),2)
+Math.pow(arrayTerbobot.get(a).get(2)-
arraySolusiIdeal.get(0).get(2),2));
72             jarakIdealNegatif =
Math.sqrt(Math.pow(arrayTerbobot.get(a).get(0)-
arraySolusiIdeal.get(1).get(0),2)
+Math.pow(arrayTerbobot.get(a).get(1)-
arraySolusiIdeal.get(1).get(1),2)
+Math.pow(arrayTerbobot.get(a).get(2)-
arraySolusiIdeal.get(1).get(2),2));
73             jumlahJarakIdeal = jarakIdealPositif +
jarakIdealNegatif;
74
75         arrayJarakIdeal.add(Arrays.asList(jarakIdealPositif,jarakIdea
lNegatif,jumlahJarakIdeal));
76     }
77     Log.d("jarak ideal", "Jarak Ideal : " +
arrayJarakIdeal);
78
79     //untuk mencari nilai prefensi dan sorting data
80     //untuk mencari nilai prefensi
81     for (int a=0; a < arrayJarakIdeal.size(); a++){
82         nilaiPrefensi = arrayJarakIdeal.get(a).get(1) /
arrayJarakIdeal.get(a).get(2);
83         arrayDataRute.get(a).set(5,
String.valueOf(nilaiPrefensi));
84
85     //arrayNilaiPrefensi.add(Arrays.asList(nilaiPrefensi));
86     }
87     Log.d("prefensi", "Nilai prefensi : " +
arrayDataRute);
88
89     //untuk sorting data

```

90	<code>Collections.sort(arrayDataRute,</code>	<code>new</code>
	<code>Comparator<List<String>>() {</code>	
91	<code> @Override</code>	
92	<code> public int compare(List<String> o1, List<String></code>	
	<code>o2) {</code>	
93	<code> return o2.get(5).compareTo(o1.get(5));</code>	
94	<code> }</code>	
95	<code>});</code>	
96		
97	<code>}</code>	
98		
99	<code>}</code>	

Berikut ini adalah penjelasan dari kode program Tabel 5.15 :

1. Baris 1

Merupakan proses inisialisasi *method* TOPSIS

2. Baris 4 – 6

Merupakan proses perhitungan pembagi TOPSIS, dimana pembagi ini akan membagi data untuk kriteria jarak, waktu tempuh dan kesulitan medan

3. Baris 8 – 16

Merupakan proses perhitungan normalisasi TOPSIS, dimana kriteria yang sebelumnya akan dibagi dengan masing – masing kolom sesuai dengan kriterianya. Dan proses ini akan terus berulang hingga pada baris terakhir dari kolom yang dimaksud. Setelah proses selesai maka hasil dari perhitungan normalisasi akan dimasukkan kedalam *arraylist* dua dimensi dengan nama *arrayNormalisasi*

4. Baris 17 – 26

Merupakan proses perhitungan nilai terbobot TOPSIS, dimana hasil dari *arrayNormalisasi* akan dikalikan dengan bobot nilai yang pengguna masukan pada *class* Kriteria sebelumnya. Dan proses ini akan terus berulang hingga pada baris terakhir dari kolom yang dimaksud. Setelah proses selesai maka hasil dari perhitungan terbobot akan dimasukkan kedalam *arraylist* dua dimensi dengan nama *arrayTerbobot*

5. Baris 29 – 34

Merupakan proses deklarasi *variable global* untuk nilai maksimum dan minimum masing – masing kriteria

6. Baris 39 – 67

Merupakan proses perulangan untuk mencari nilai maksimal dan nilai minimal dari setiap kolom jarak, waktu tempuh dan kesulitan medan. Setelah berhasil mendapatkan nilai maksimal dan nilai minimal dari masing – masing kriteria maka selanjutnya nilai maksimal dan nilai

minimal akan di simpan dalam arraylist dua dimensi dengan nama arraySolusiIdeal

7. Baris 70 – 78

Merupakan proses mencari jarak ideal positif dan negatif dari kalkulasi TOPSIS, dimana setelah proses kalkulasi selesai maka hasil kalkulasi akan disimpan dalam arraylist dua dimensi dengan nama arrayJarakIdeal

8. Baris 81 – 87

Merupakan proses perhitungan nilai prefensi TOPSIS

9. Baris 90 – 95

Merupakan proses mengurutkan nilai prefensi dari nilai yang terbesar ke nilai yang terkecil, dimana menggunakan *library android studio compare* untuk proses pengurutannya.

5.6 Implementasi Antarmuka Sistem

Implementasi antarmuka sistem bersumber dari perancangan antarmuka sistem yang telah dijelaskan penulis pada bab 4 sebelumnya. Adapun isi dari implementasi antarmuka sistem dapat dilihat pada Gambar 5.2 sampai dengan Gambar 5.6 :



Gambar 5.2 Implementasi Antarmuka Beranda

Gambar 5.2 ialah implementasi antarmuka sistem beranda. Dimana saat aplikasi dijalankan maka pertama kali pengguna akan melihat tampilan ini



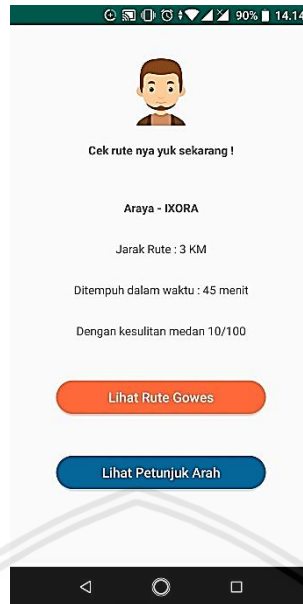
Gambar 5.3 Implementasi Antarmuka Fungsionalitas Memberikan Bobot Nilai

Gambar 5.3 ialah implementasi antarmuka sistem untuk fungsionalitas memberikan bobot nilai. Dimana setelah pengguna menekan tombol mulai gowes pada tampilan beranda, maka pengguna bisa menentukan bobot nilai untuk masing – masing kriteria



Gambar 5.4 Implementasi Antarmuka Fungsionalitas Melihat Rekomendasi Rute Gowes

Gambar 5.4 ialah implementasi antarmuka sistem untuk fungsionalitas melihat rekomendasi rute gowes. Dimana setelah pengguna memberikan bobot nilai untuk masing – masing kriteria maka aplikasi akan menampilkan hasil rute dari perhitungan TOPSIS



Gambar 5.5 Implementasi Antarmuka Fungsionalitas Melihat Detail Rute

Gambar 5.5 ialah implementasi antarmuka sistem untuk fungsionalitas melihat detail rekomendasi. Setelah pengguna mendapatkan hasil rekomendasi maka pengguna bisa memilih salah satu rekomendasi dan aplikasi akan menampilkan detail rekomendasi tersebut

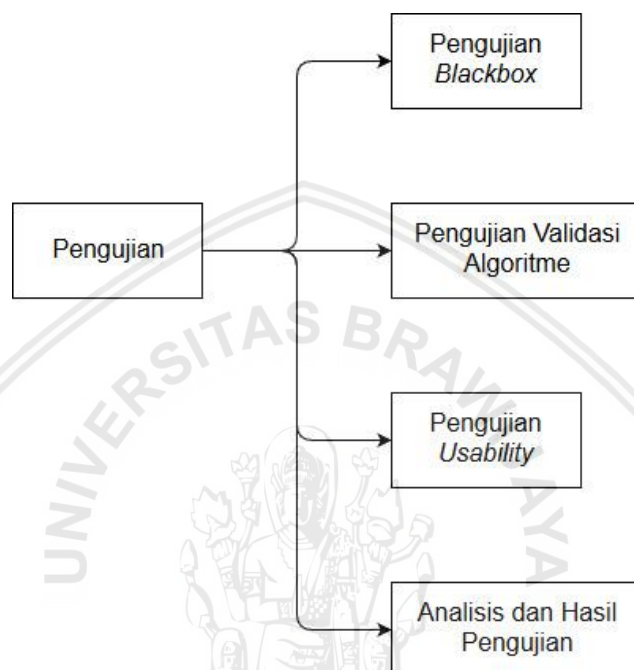


Gambar 5.6 Implementasi Antarmuka Fungsionalitas Melihat Petunjuk Arah

Gambar 5.6 ialah implementasi antarmuka sistem untuk fungsionalitas melihat petunjuk arah. Pada tampilan detail rekomendasi, pengguna bisa melihat petunjuk arah ke titik awal bersepeda dengan menekan salah satu tombol dan secara otomatis aplikasi akan membuka *Google Maps* untuk melihat rute tersebut

BAB 6 PENGUJIAN

Bab pengujian menjabarkan mengenai mekanisme pengujian yang diterapkan dalam menguji sistem rekomendasi rute gowes di kota Malang. Dalam pengujian sendiri, penulis membagi menjadi tiga tahapan yaitu pengujian *blackbox*, pengujian validasi algoritme dan pengujian *usability*. Diagram alir pengujian sistem rekomendasi rute gowes di kota Malang dapat dilihat pada Gambar 6.1 :



Gambar 6.1 Diagram Alir Pengujian

6.1 Pengujian *Blackbox*

Pengujian *blackbox* ialah pengujian sistem yang berpusat pada fungsionalitas dari sistem itu sendiri. Pengujian sistem yang dilakukan oleh penulis berlandaskan pada *usecase scenario* yang sudah dijabarkan pada bab 4 sebelumnya. Pengujian *blackbox* sistem rekomendasi rute gowes di kota Malang dapat dilihat pada Tabel 6.1 sampai dengan Tabel 6.4 :

Tabel 6.1 Pengujian Fungsionalitas Memberikan Bobot Nilai Untuk Setiap Kriteria

Nama fungsionalitas	Memberikan bobot nilai untuk setiap kriteria
Tujuan pengujian	Untuk meyakinkan pengguna mampu memberikan nilai pada masing – masing kriteria rute gowes
Langkah utama pengujian	1. Pengguna membuka aplikasi 2. Pengguna menekan tombol mulai gowes

	3. Pengguna memberikan nilai pada masing – masing kriteria 4. Pengguna menekan tombol lihat rute gowes
Langkah alternatif pengujian	-
Hasil yang diharapkan	Sistem mampu menyajikan halaman kriteria dan mampu menyimpan bobot nilai untuk masing – masing kriteria
Hasil keluaran sistem	Sistem mampu menyajikan halaman kriteria dan mampu menyimpan bobot nilai untuk masing – masing kriteria seperti dibawah ini : 
Kesimpulan	Valid

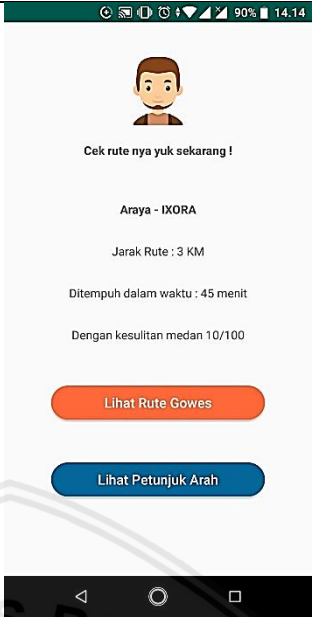
Tabel 6.2 Pengujian Fungsionalitas Melihat Rekomendasi Rute Gowes

Nama fungsionalitas	Melihat Rekomendasi Rute Gowes
Tujuan pengujian	Untuk meyakinkan pengguna mampu mengetahui hasil rekomendasi rute gowes sesuai dengan bobot nilai yang telah diberikan sebelumnya
Langkah utama pengujian	1. Pengguna telah menekan tombol lihat rute gowes 2. Aplikasi menampilkan hasil rekomendasi rute gowes 3. Pengguna memilih salah satu rute yang telah direkomendasikan

Langkah alternatif pengujian	-
Hasil yang diharapkan	Sistem dapat menampilkan rekomendasi rute gowes pada halaman daftar rute
Hasil keluaran sistem	Sistem dapat menampilkan rekomendasi rute seperti dibawah ini : 
Kesimpulan	Valid

Tabel 6.3 Pengujian Fungsionalitas Melihat Detail Rekomendasi Rute

Nama fungsionalitas	Melihat Detail Rekomendasi Rute
Tujuan pengujian	Untuk meyakinkan pengguna mampu mengetahui detail dari rekomendasi rute yang dipilih
Langkah utama pengujian	1. Pengguna menetapkan salah satu rekomendasi rute 2. Aplikasi menyajikan detail dari rekomendasi rute yang dipilih
Langkah alternatif pengujian	-
Hasil yang diharapkan	Sistem mampu menyajikan detail rekomendasi rute dengan data – data seperti nama rute, jarak rute, waktu tempuh rute, dan kesulitan medan
Hasil keluaran sistem	Sistem mampu menyajikan detail rekomendasi rute dengan data – data seperti nama rute, jarak rute, waktu tempuh rute, dan kesulitan medan seperti dibawah ini :

	
Kesimpulan	Valid

Tabel 6.4 Pengujian Fungsionalitas Melihat Petunjuk Arah

Nama fungsionalitas	Melihat Petunjuk Arah
Tujuan pengujian	Untuk meyakinkan pengguna mampu mengetahui petunjuk arah rute gowes yang dipilih melalui sistem
Langkah utama pengujian	1. Pengguna berada pada halaman detail rekomendasi rute 2. Pengguna menekan tombol lihat rute gowes 3. Aplikasi akan membuka <i>Google Maps</i>
Langkah alternatif pengujian	1. Pengguna berada pada halaman detail rekomendasi rute 2. Pengguna menekan tombol lihat petunjuk arah 3. Aplikasi akan membuka <i>Google Maps</i>
Hasil yang diharapkan	Sistem mampu menyajikan petunjuk arah rute yang dipilih pengguna
Hasil keluaran sistem	Sistem mampu menyajikan petunjuk arah rute yang dipilih pengguna seperti dibawah ini :

	
Kesimpulan	Valid

6.2 Pengujian Validasi Algoritme

Pengujian validasi algoritme merupakan pengujian yang dilakukan dengan cara mencocokkan hasil kalkulasi manual dengan kalkulasi dari keluaran sistem. Pada pengujian ini, penulis memberikan bobot nilai yang sama pada sistem dan

juga pada perhitungan manual. Adapun bobot nilai yang penulis uji dapat dilihat pada Tabel 6.5 :

Tabel 6.5 Bobot Nilai Pengujian Validasi Algoritme

Kriteria	Bobot Nilai
Jarak (Km)	0.5
Waktu Tempuh (Menit)	0.3
Kesulitan Medan (0-1)	0.2

Untuk pengujian ini, penulis menggunakan data rute gowes yang berasal dari *Firestore* dengan rincian data yaitu jarak rute, waktu tempuh rute dan kesulitan medan rute. Data rute gowes sistem rekomendasi rute gowes di kota Malang dapat dilihat pada Tabel 6.6 :

Tabel 6.6 Data Rute Gowes

No	Nama Rute	Jarak (Km)	Waktu Tempuh (Menit)	Kesulitan Medan (0-100)
1	Araya - Alun alun Batu	22	60	75
2	Rampal - Alun alun Malang	3	15	30
3	Araya - IXORA	5	20	10
4	Rampal - Bukit Tidar	12	45	80
5	Alun alun Malang - SMAN 10 Nala Taruna	10	30	70
6	Rampal - Coban Jidor	23	60	80
7	Idjen - Alun alun Malang	2	10	20
8	Araya - Pasar Besar	7	25	25
9	Araya - Tumpang	17	45	20

Setelah penulis melaksanakan pengujian validasi algoritme pada sistem rekomendasi rute gowes di kota Malang, berikut ialah perbandingan hasil perhitungan sistem dengan hasil kalkulasi manual yang dijabarkan pada Tabel 6.7 :

Tabel 6.7 Perbandingan Perhitungan Sistem dengan Perhitungan Manual

No	Perhitungan Sistem		Perhitungan Manual		Kesimpulan
	Nama Rute	Nilai Prefensi	Nama Rute	Nilai Prefensi	
1	Araya - Alun alun Batu	0.954500	Araya - Alun alun Batu	0.954501	Valid
2	Rampal - Alun alun Malang	0.098619	Rampal - Alun alun Malang	0.098619	Valid
3	Araya - IXORA	0.147883	Araya - IXORA	0.147884	Valid
4	Rampal - Bukit Tidar	0.554762	Rampal - Bukit Tidar	0.554763	Valid
5	Alun alun Malang - SMAN 10 Nala Taruna	0.42885	Alun alun Malang - SMAN 10 Nala Taruna	0.42885	Valid
6	Rampal - Coban Jidor	1	Rampal - Coban Jidor	1	Valid
7	Idjen - Alun alun Malang	0.039936	Idjen - Alun alun Malang	0.039936	Valid
8	Araya - Pasar Besar	0.248248	Araya - Pasar Besar	0.248248	Valid
9	Araya - Tumpang	0.648147	Araya - Tumpang	0.648147	Valid

6.3 Pengujian *Usability*

Pengujian *usability* ini menggunakan metode *system usability scale* dimana dalam pengujian ini, penulis mengumpulkan 5 responden untuk menjawab pertanyaan – pertanyaan dengan bobot nilai tertentu. Terdapat 10 pertanyaan dalam *system usability scale* ini dan penulis sudah memberikan bobot nilai dengan nilai minimum 1 dan nilai maksimum 5. Dalam penilaian ini, penulis tidak memberikan nilai secara langsung namun menggunakan skala antara lain sangat tidak setuju, tidak setuju, netral, setuju dan sangat setuju yang mewakili nilai terendah sampai dengan nilai terbesar. Karakteristik dari responden dalam pengujian *usability* ini meliputi jenis kelamin, domisili, serta usia. Jenis kelamin

responden dalam penelitian ini ialah laki – laki. Domisili setiap responden yakni Kota Malang dan rentang usia responden ialah berusia antara 15 – 22 tahun. Adapun pertanyaan – pertanyaan serta jumlah responden yang menjawab dapat dilihat pada Tabel 6.8 :

Tabel 6.8 Sebaran Jawaban Responden Terhadap Kuisisioner

No	Pertanyaan	Jumlah				
		STS	TS	N	S	SS
1	Saya berfikir akan menggunakan aplikasi ini				3	2
2	Menurut saya,aplikasi ini tidak harus dibuat serumit ini		3	2		
3	Menurut saya,aplikasi ini mudah untuk digunakan					5
4	Saya membutuhkan bantuan orang lain dalam menggunakan aplikasi ini	3	1	1		
5	Saya menemukan berbagai fungsi di aplikasi ini diintegrasikan dengan baik				4	1
6	Menurut saya,banyak ketidaksesuaian dalam aplikasi ini	1	3	1		
7	Menurut saya,kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan cepat				1	4
8	Menurut saya,aplikasi ini sangat rumit untuk digunakan	4	1			
9	Saya merasa sangat percaya diri untuk menggunakan aplikasi ini				3	2
10	Saya perlu belajar banyak hal sebelum menggunakan aplikasi ini	1	4			

Setelah mengetahui detail dari masing – masing pertanyaan,selanjutnya penulis akan memaparkan penilaian masing – masing responden. Nantinya hasil penilaian responden akan dikalikan dengan nilai 2,5 agar hasil dari penilaian responden dapat mencapai nilai maksimal 100. Hasil dari penilaian responden bisa dilihat pada Tabel 6.9 :

Tabel 6.9 Hasil Penilaian Responden

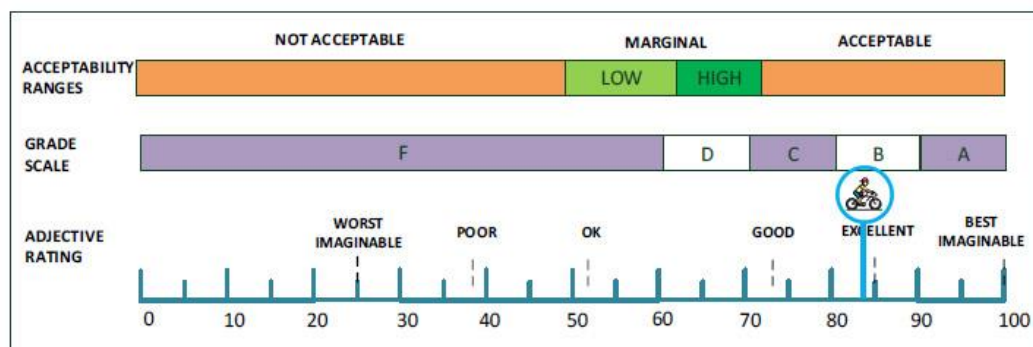
Responden	Penilaian	Nilai Akhir
1	$30 \times 2,5$	75
2	$36 \times 2,5$	90
3	$36 \times 2,5$	90
4	$35 \times 2,5$	87,5
5	$32 \times 2,5$	80
Rata - rata		$422,5 / 5 = 84,5$

6.4 Analisis dan Hasil Pengujian

Pada sub bab ini, penulis menjabarkan analisis dari hasil pengujian yang telah penulis lakukan sebelumnya. Pada pengujian *blackbox* diatas dapat diambil kesimpulan bahwa seluruh fungsionalitas yang terdapat pada sistem rekomendasi rute gowes di kota malang beroperasi dengan baik serta cocok dengan keluaran yang diinginkan oleh penulis. Dari ke empat fungsionalitas yang diuji, seluruhnya menghasilkan nilai valid.

Pada pengujian validasi algoritme, penulis telah melakukan pengujian dengan membandingkan nilai prefensi dari perhitungan manual dengan nilai prefensi dari perhitungan sistem. Dari ke sembilan rute yang telah dihitung nilai prefensi menggunakan perhitungan manual mengeluarkan nilai yang sama dengan nilai prefensi yang dikeluarkan oleh sistem sehingga dapat disimpulkan bahwa algoritme yang diterapkan pada sistem rekomendasi rute gowes adalah valid. Sehingga hasil rekomendasi dengan cara menghitung manual adalah sama dengan hasil rekomendasi perhitungan sistem.

Sementara pada pengujian *usability*, didapatkan nilai akhir atau nilai rata – rata dari seluruh penilaian yang dilakukan oleh responden sama dengan 84,5. Selanjutnya, hasil rata – rata yang didapat akan dimasukan kedalam skala *system usability scale* dimana terdapat dua penilaian terhadap suatu aplikasi yaitu melalui *acceptability rangers* dan *grade scale* . Hasil dari pengujian usability ini didapatkan bahwa aplikasi sistem rekomendasi rute gowes mendapatkan nilai 84,5 dengan *grade scale* B dan mendapatkan nilai *acceptable* pada *acceptability rangers*. Untuk melihat penilaian tersebut, penulis akan memaparkannya pada Gambar 6.6 :



Gambar 6.6 Penentuan Hasil Penilaian

Sumber : Ependi, Panjaitan , dan Hutrianto (2017)



BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berlandaskan observasi yang telah dilakukan oleh penulis, maka bisa ditarik kesimpulan sebagai berikut :

1. Sistem rekomendasi rute gowes di kota Malang dirancang dengan menerapkan metode algoritme TOPSIS. Dalam langkah pengembangan, sistem ini terlebih dahulu memiliki beberapa perancangan diantaranya adalah perancangan *use case diagram*, *use case scenario*, *activity diagram*, *sequence diagram*, *class diagram*, perancangan basis data, perancangan antarmuka serta perancangan algoritme TOPSIS itu sendiri. Sistem rekomendasi rute gowes di kota Malang ini memiliki kriteria berupa jarak rute, waktu tempuh rute serta kesulitan medan.
2. Sistem rekomendasi rute gowes di kota Malang diimplementasikan dengan memanfaatkan *Android Studio*, *Google Firebase*, serta algoritme TOPSIS untuk mendapatkan hasil rekomendasi rute yang sesuai. Algoritme TOPSIS sendiri akan menghasilkan nilai prefensi yang nantinya akan dilakukan pengurutan dengan nilai prefensi terbesar ke nilai prefensi terkecil dari setiap rute yang telah ada.
3. Sistem rekomendasi rute gowes di kota Malang memiliki hasil pengujian *usability* dengan nilai akhir 84,5 atau menghasilkan nilai *adjective rating* yaitu *good* dengan *grade scale* bernilai B. Selain itu pengujian validasi algoritme menghasilkan nilai prefensi antara keluaran sistem sama dengan keluaran perhitungan manual. Untuk pengujian *blackbox* yang telah dilakukan, seluruh fungsionalitas yang telah diuji berhasil mendapatkan kesimpulan valid.

7.2 Saran

Saran yang mampu penulis sampaikan untuk penelitian selanjutnya yaitu sebagai berikut :

1. Perlunya pengoptimalan penggunaan *Google Maps* pada aplikasi ini sehingga nantinya pengguna bisa langsung melihat rute gowes tanpa harus berpindah aplikasi.
2. Pengembangan selanjutnya diharapkan memperluas cakupan wilayah rute sepeda ke beberapa daerah yang dekat dengan kota Malang seperti kabupaten Malang maupun kota Batu dikarenakan terbatasnya rute yang ada di kota Malang.

DAFTAR REFERENSI

- Avelina, Soraya Rahma., 2018. *APLIKASI DELIVERY ORDER MINIMARKET MENGGUNAKAN FASILITAS GPS TRACKING ANDROID*. S1. Universitas Muhammadiyah Sidoarjo. Tersedia di: <http://eprints.umsida.ac.id/1818/> [Diakses 27 Februari 2019]
- Developer, 2019. Developer Android. [online] Tersedia di: <https://developer.android.com/guide/platform?hl=id> [Diakses 24 Februari 2019].
- Dharmawan, Alexander., Ismail, Harries Arizonia., 2015. *APLIKASI PENGENALAN BUDAYA INDONESIA PADA SEKOLAH DASAR KRISTEN 3 YSKI BERBASIS ANDROID*, [e-journal]. Tersedia melalui: Komputaki <http://www.unaki.ac.id/ejournal/index.php/komputaki/article/view/147/159> [Diakses 24 Februari 2019].
- Effendi, KAS., Santoso, Edy., Hidayat, Nurul. 2018. Implementasi Metode TOPSIS Untuk Penentuan Finalis Duta Wisata Joko Roro Kabupaten Malang, [e-journal]. Tersedia melalui: JPTIHK <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/855> [Diakses 10 Januari 2019].
- Ependi, Usman., Panjaitan, Febriyanti., Hutrianto., 2017. System Usability Scale Antarmuka Palembang Guide Sebagai Media Pendukung Asian Games XVIII, [e-journal]. Tersedia melalui : Journal of Information Systems Engineering and Business Intelligence <https://e-journal.unair.ac.id/JISEBI/article/view/4118/3886> [Diakses 18 Februari 2019].
- Firebase, 2019. Firebase Branding. [online] Tersedia di: <https://firebase.google.com/downloads/brand-guidelines/PNG/logo-standard.png?hl=id> [Diakses 27 Februari 2019].
- Firebase, 2019. Firebase Dokumentasi. [online] Tersedia di: <https://firebase.google.com/docs/> [Diakses 10 Januari 2019].
- Google, 2019. Google Maps. [online] Tersedia di: <https://www.google.com/maps/dir/Jl.+Mayjend.+Panjaitan,+Penanggungan,+Klojen,+Kota+Malang,+Jawa+Timur,+Indonesia/Ngalup+Coworking+Space,+Jl.+Sudimoro,+Mojolangu,+Lowokwaru,+Malang+City,+East+Java+65142/@-7.9430797,112.6127728,3275m/data=!3m1!1e3!4m14!4m13!1m5!1m1!1s0x2dd6282ad44d154b:0x9fbc7c0cbcbc5329!2m2!1d112.6212501!2d-7.9554931!1m5!1m1!1s0x2dd629fb35c9b5ab:0xd0b3301c0563c39b!2m2!1d112.6274664!2d-7.9309211!3e0> [Diakses 27 Februari 2019].
- Google, 2019. Google Maps Dokumentasi. [online] Tersedia di: <https://developers.google.com/maps/documentation/android-sdk/intro> [Diakses 10 Januari 2019].

- Karuniawati, Suri., Hakim, Iman Lukmanul., 2015. Implementasi Metode Cause Effect Graphing (CEG) dalam Pengujian Requirement Perangkat Lunak (Studi Kasus: Aplikasi G-College), [e-journal]. Tersedia melalui: Telkomuniversity <<https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/2967/2814>>[Diakses 10 Januari 2019].
- Katadata., 2016. Pengguna Smartphone di Indonesia 2016-2019. [online] Tersedia di: <<https://databoks.katadata.co.id/datapublish/2016/08/08/pengguna-smartphone-di-indonesia-2016-2019>>[Diakses 12 Februari 2019]
- Kemdikbud., 2016. KBBI Daring. [online] Tersedia di: <<https://kbbi.kemdikbud.go.id/entri/gowes>>[Diakses 18 Desember 2018]
- Kementrian Kesehatan Republik Indonesia, 2017. Olahraga & 59 Manfaat Bagi Tubuh 2017. [Online] Available at: <<http://www.depkes.go.id/development/site/depkes/index.php?cid=1-17042500002&id=olahraga-59-manfaat-bagi-tubuh.html>>[Diakses 18 Desember 2018].
- Kusuma, DH., Shodiq, MN., 2017. Sistem Rekomendasi Destinasi Pariwisata Menggunakan Metode Hibrid Case Based Reasoning dan Location Based Service Sebagai Pemandu Wisatawan di Banyuwangi, [e-journal]. Tersedia melalui: INTENSIF <<http://ojs.unpkediri.ac.id/index.php/intensif/article/view/540>>[Diakses 10 Januari 2019].
- Lemke, Gillian., 2018. *The Software Development Life Cycle And Its Application*. S2. Eastern Michigan University. Tersedia di<<https://commons.emich.edu/honors/589/>>[Diakses 24 Februari 2019]
- Lengkong, Hendra Nugraha., Sinsuw, Alicia AE., Lumenta, Arie SM., 2015. Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi Mobile GIS Berbasis Android Yang Terintegrasi Pada Google Maps, [e-journal]. Tersedia melalui: Jurnal Teknik Elektro dan Komputer UNSRAT <<https://ejournal.unsrat.ac.id/index.php/elekdankom/article/view/6817/6341>>[Diakses 10 Januari 2019].
- Mallu, Satriawaty., 2015. SISTEM PENDUKUNG KEPUTUSAN PENENTUAN KARYAWAN KONTRAK MENJADI KARYAWAN TETAP MENGGUNAKAN METODE TOPSIS, [e-journal]. Tersedia melalui: JITTER <<http://jitter.widyatama.ac.id/index.php/jitter/article/view/32>>[Diakses 10 Januari 2019].
- Maryam, Amrah., Siddiqui, Nadia., Qadeer, MA., dan Umar, MS., 2016. Travel Management System using GPS & Geo Tagging on Android Platform, [e-journal]. Tersedia melalui: IEEEExplore <<https://ieeexplore.ieee.org/abstract/document/7759019>>[Diakses 9 Januari 2019].
- Mimin, 2016. APLIKASI PELAKSANAAN PEMBELAJARAN AKADEMIK AR-RAFI' MODUL UJI KOMPETENSI SISWA TERTULIS DAN MANAJEMEN RAPAT KERJA

- (Studi Kasus: SD Ar-rafi'), [e-journal]. Tersedia melalui: Telkomuniversity <<https://libraryproceeding.telkomuniversity.ac.id/index.php/appliedscience/article/view/3627/3440>>[Diakses 10 Januari 2019].
- Oktilas, Ahmad Fali., Siswanti, Sri Desy., Rachman, M Dieka., 2015. Akurasi Pembacaan GPS pada Android untuk Location Based Service (Studi Kasus: Informasi Lokasi SMA di Palembang), [e-journal]. Tersedia melalui: Ilmu Komputer Agri-Informtika <<http://journal.ipb.ac.id/index.php/jika/article/view/14569/10795>>[Diakses 18 Februari 2019].
- Palar, Chrisly M., Wongkar, Djon., Ticoalu, Shane HR., 2015. Manfaat Latihan Olahraga Aerobik Terhadap Kebugaran Fisik Manusia, [e-journal]. Tersedia melalui: Perhimpunan Ahli Anatomi Indonesia UNSRAT <<https://ejournal.unsrat.ac.id/index.php/ebiomedik/article/view/7127/6638>>[Diakses 18 Februari 2019].
- Pemerintah Kota Malang., 2015. Kota Malang Bertekad Tambah Medali di Cabor Balap Sepeda. [online] Tersedia di: <<https://malangkota.go.id/2015/06/09/kota-malang-bertekad-tambah-medali-di-cabor-balap-sepeda/>>[Diakses 18 Februari 2019]
- Pemerintah Kota Malang., 2016. Ratusan Atlet Sepeda Berkompetisi di Piala Wali Kota. [online] Tersedia di: <<https://malangkota.go.id/2016/10/28/ratusan-atlet-sepeda-berkompetisi-di-piala-wali-kota/>>[Diakses 18 Februari 2019]
- Pemerintah Kota Malang., 2019. Atlet Balap Sepeda Kota Malang Siap Tampil di ACC 2019. [online] Tersedia di: <<https://malangkota.go.id/2019/01/07/atlet-balap-sepeda-kota-malang-siap-tampil-di-acc-2019/>>[Diakses 18 Februari 2019]
- Pemerintah Kota Malang., 2019. Sejarah Malang. [online] Tersedia di: <<https://malangkota.go.id/sekilas-malang/sejarah-malang/>>[Diakses 18 Februari 2019]
- Putra, DMDU., Pratama, IPA., 2016. Rancang Bangun Sistem Pendukung Keputusan Penentuan Penerima Beasiswa Dengan Menggunakan Metode TOPSIS, [e-journal]. Tersedia melalui: sacies <<https://jurnal.stiki-indonesia.ac.id/index.php/sacies/article/view/114/9>>[Diakses 10 Januari 2019].
- Statcounter., 2019. Mobile Operating System Market Share Indonesia. [online] Tersedia di: <<http://gs.statcounter.com/os-market-share/mobile/indonesia/#monthly-201801-201901-bar>>[Diakses 18 Februari 2019]
- Supraja, S., Kousalya, P., 2016. A Comparative Study by AHP and TOPSIS for the Selection of All Round Excellence Award, [e-journal]. Tersedia melalui: IEEEExplore <<https://ieeexplore.ieee.org/document/7755271>>[Diakses 10 Januari 2019].

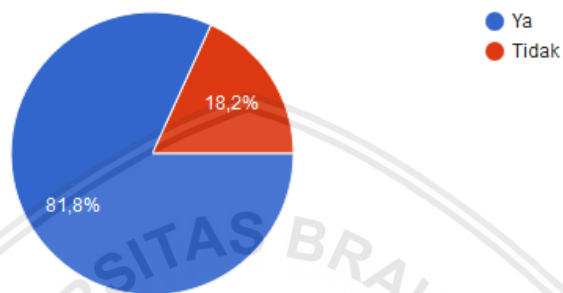
- Wahyo, Bambang Tri., Anggriawan, Angga Widya., 2015. Sistem Rekomendasi Paket Wisata Se-malang Raya Menggunakan Metode Hybrid Content Based Dan Collaborative, [e-journal]. Tersedia melalui: Jurnal Ilmiah Teknologi Informasi Asia <<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/96/77>>[Diakses 18 Februari 2019].
- Windarto, Agus Perdana., 2017. IMPLEMENTASI METODE TOPSIS DAN SAW DALAM MEMBERIKAN REWARD PELANGGAN, [e-journal]. Tersedia melalui: Google Scholar <<https://scholar.google.co.id>>[Diakses 9 Januari 2019].



LAMPIRAN A DATA SURVEI PEMINAT GOWES DI KOTA MALANG

Apakah selama ini anda mengalami kesulitan dalam mencari rute bersepeda di kota Malang ?

11 tanggapan



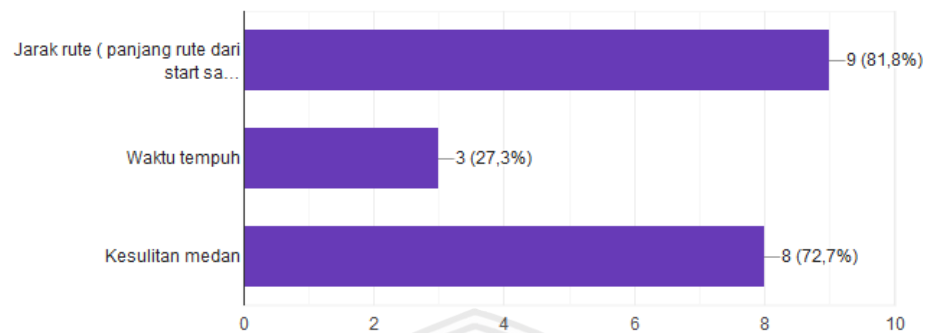
Jika ya,kesulitan apa yang anda alami ?

11 tanggapan

-
Tidak mendapatkan informasi yang sesuai
Tidak mengetahui tujuan dan jalur sepeda di kota Malang
Kurang mengetahui lamanya waktu tempuh dan medan yang ditempuh seperti apa
Penentuan rute dan jaraknya
karena jarang teman yang menyukai gowes
Karena jarang teman yang suka gowes
karena bukan orang asli malang, jadi belum tau rute bersepeda
mencari rute bersepeda yang terdekat dengan posisi saya
takut rame motor

Kriteria apa saja yang anda gunakan ketika menentukan rute bersepeda ?

11 tanggapan



```

ute
String
id...:String
...:String
g
g
g
g
ing
...
nama_ru...:String, kesulatan_med...:String, jarak:String, waktu_temp...:String, id:String, lataw...:String, longaw...:String
()
...:String
stow...:String):void
(!):String
(longaw...:String):void
(!):String
latakhir:String):void
(!):String
(longak...:String):void
g
ng):void
(!):String
(nama_ru...:String):void
me...():String
me... (kesulatan_med...:String):void
String
ak:String):void
mp...():String
mp... (waktu_temp...:String):void

```



LAMPIRAN C DATA RUTE GOWES DI KOTA MALANG

No	Nama Rute	Jarak	Waktu Tempuh	Kesulitan Medan	Latitu de Awal	Longi tude Awal	Latitu de Akhir	Longitu de Akhir
1	Araya - Alun alun Batu	22	60	75	- 7.936 5517	112.6 48122 2	- 7.8711 591	112.524 7
2	Ramp al - Alun alun Malan g	3	15	30	- 7.973 7781	112.6 38159 6	- 7.9825 193	112.628 6
3	Araya - IXORA	5	20	10	- 7.936 5517	112.6 48122 2	- 7.9352 79	112.681 7
4	Ramp al - Bukit Tidar	12	45	80	- 7.973 7781	112.6 38159 6	- 7.9442 05	112.582 1
5	Alun alun Malan g - SMAN 10 Nala Tarun a	10	30	70	- 7.982 5193	112.6 28612 6	- 8.0385 279	112.652 5
6	Ramp al - Coban Jidor	23	60	80	- 7.973 7781	112.6 38159 6	- 7.9967 932	112.802 4
7	Idjen - Alun alun Malan g	2	10	20	- 7.975 548	112.6 20709	- 7.9825 193	112.628 6
8	Araya - Pasar Besar	7	25	25	- 7.936 5517	112.6 48122 2	- 7.9879 801	112.631 8

9	Araya - Tump ang	17	45	20	- 7.936 5517	112.6 48122 2	- 8.0034 972	112.757 4
---	---------------------------	----	----	----	--------------------	---------------------	--------------------	--------------

