

Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Ricky Irfandi
NIM: 155150201111274



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

PENGESAHAN

IMPLEMENTASI TOPSIS PADA SISTEM REKOMENDASI TEMPAT WISATA DALAM KOTA MALANG BERBASIS LOKASI

SKRIPSI

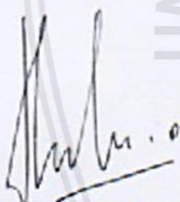
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ricky Irfandi
NIM: 155150201111274

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Juli 2019

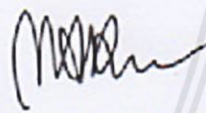
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



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

Dosen Pembimbing 2



Drs. Marji, M.T.
NIP: 19670801 199203 1 001

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, 2 Juli 2019



Ricky Irfandi

NIM: 155150201111274

PRAKATA

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena hanya dengan rahmat dan karunia-Nya, skripsi yang berjudul “Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi” dapat penulis selesaikan. Penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada pihak pihak yang telah membantu dan mendukung selama penulisan skripsi ini berlangsung, diantaranya :

1. Ibu Ratih Kartika Dewi, S.T., M.Kom, selaku dosen pembimbing 1 yang telah memberikan masukan, ilmu, dan saran dalam menyelesaikan skripsi ini.
2. Bapak Drs. Marji, M.T, selaku dosen pembimbing 2 yang telah memberikan masukan, ilmu, dan saran dalam menyelesaikan skripsi ini.
3. Ayah Trisno Sugeng Riadi dan Ibu Heni Supriatin, S.H beserta keluarga besar yang telah memberikan doa, nasihat, serta dukungan yang tiada hentinya hingga skripsi ini selesai
4. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya yang telah membagi ilmunya kepada penulis.
5. Teman-teman Informatika 2015 yang selalu memberi semangat dan membantu dalam pengerjaan skripsi ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat secara langsung atau tidak langsung sehingga terselesaikannya skripsi ini.

Penulis sadar bahwa skripsi ini masih terdapat banyak kekurangan. Kritik dan saran yang bersifat membangun diharapkan. Semoga skripsi ini dapat membawa manfaat bagi pihak lain yang menggunakannya.

Malang, 2 Juli 2019

Penulis

rickyirfandi@gmail.com

ABSTRAK

Ricky Irfandi, Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi.

Pembimbing: Ratih Kartika Dewi, S.T., M.Kom. dan Drs. Marji, M.T.

Kota Malang merupakan kota kedua terbesar di Jawa Timur yang dikenal dengan julukan kota pariwisata karena memiliki banyak destinasi wisata yang menarik. Namun, banyaknya pilihan destinasi wisata di Kota Malang membuat wisatawan menjadi bingung memilih wisata yang cocok sesuai dengan keinginannya karena keterbatasan dana dan waktu. Untuk mengatasi permasalahan tersebut, diperlukan sebuah aplikasi berbasis *mobile* pendukung keputusan yang dapat memberikan rekomendasi wisata sehingga diharapkan wisatawan mendapatkan rekomendasi wisata yang sesuai dengan kriteria yang diinginkan darimana saja dan kapan saja.

Pengembangan sistem pada penelitian ini dilakukan dengan berbasis Android sehingga diharapkan dapat digunakan oleh pengguna yang luas. Sistem dibangun dengan menggunakan metode TOPSIS. Metode TOPSIS ini digunakan untuk melakukan ranking rekomendasi wisata sesuai dengan kriteria yang dimasukkan oleh pengguna berdasarkan data obyek wisata yang disediakan oleh *web services*. Sensor GPS digunakan untuk mengetahui lokasi pengguna saat ini yang akan digunakan untuk proses perhitungan. Selain itu, pengguna dapat melakukan navigasi pada halaman rekomendasi dengan menekan tombol navigasi pada wisata yang dipilih. Nantinya aplikasi akan membuka Google Maps dan melakukan navigasi dari lokasi pengguna saat ini menuju wisata yang dipilih.

Sistem yang dibangun berhasil memenuhi kebutuhan fungsional dengan nilai valid 100% pada pengujian fungsional. Selain itu pengujian validasi algoritme yang dilakukan dengan membandingkan hasil rekomendasi dengan perhitungan manual dan perhitungan sistem memperoleh kecocokan 100%. Dan untuk pengujian akurasi oleh pakar mendapatkan nilai akurasi sebesar 83,33%.

Kata kunci: Sistem Pendukung Keputusan, wisata, TOPSIS, Malang

ABSTRACT

Ricky Irfandi, Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi.

Pembimbing: Ratih Kartika Dewi, S.T., M.Kom. dan Drs. Marji, M.T.

Malang is the second largest city in East Java, which is known as the tourism city because it has many interesting tourist destinations. However, many choices of tourist destinations in Malang confuse the tourists about choosing suitable tourism according to their wishes because of the limited funds and time. To overcome this problem, a mobile-based application is needed for decision-making that can provide tourist recommendations, so that tourists are expected to get travel recommendations that match their desired criteria from anywhere and anytime.

The system developed in this study was carried out based on Android so that it was expected to be used by a wide range of users. The system was built using the TOPSIS method. The TOPSIS method was used to rank tourist recommendations according to the criteria entered by users based on tourist data provided by web services. The GPS sensor was used to determine the current user's location to be used in the calculation process. In addition, users could navigate to the recommendation page by pressing the navigation button on the selected tour. Later the application would lead to open Google Maps and navigate from the location of the current user to the selected tourist destination.

The system built has successfully fulfilled functional requirements with a valid value of 100% on functional testing. Besides, the validation testing of algorithms which is done by comparing the results of recommendations with manual calculations and the calculation of the system gets a 100% match. And for accuracy testing by experts gets an accuracy value of 83.33%.

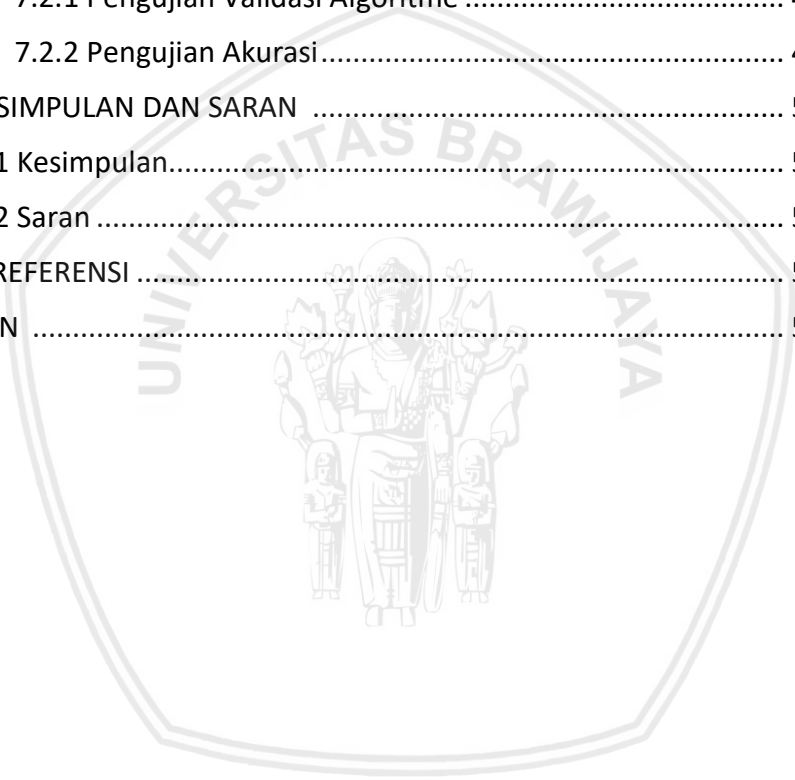
Keywords : Decision Support System, Tourism, TOPSIS, Malang

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Sistem Pendukung Keputusan	5
2.3 <i>Technique For Order Preference by Similiarity (TOPSIS)</i>	5
2.4 <i>Haversine Formula</i>	6
2.5 <i>Global Positioning System (GPS)</i>	7
2.6 Android	7
2.7 Android Studio	7
2.8 <i>JavaScript Object Notation (JSON)</i>	7
2.9 Web Service	8
BAB 3 METODOLOGI	9
3.1 Strategi Penelitian.....	9
3.2 Subjek Penelitian	9
3.3 Lokasi Penelitian	9

3.4 Metode Pengumpulan Data.....	9
3.5 Peralatan Pendukung.....	9
3.6 Metode Pengembangan Sistem.....	10
BAB 4 ANALISIS KEBUTUHAN	11
4.1 Gambaran Aplikasi	11
4.2 Identifikasi Aktor.....	12
4.3 Analisis Kebutuhan Fungsional	13
4.3.1 Use Case Diagram.....	13
4.3.2 Spesifikasi Use Case	13
4.3.3 Skenario Use Case	13
4.4 Analisis Kebutuhan Fungsional	15
BAB 5 PERANCANGAN PERANGKAT LUNAK	16
5.1 Perancangan Activity Diagram.....	16
5.2 Perancangan Sequence Diagram	18
5.3 Perancangan Class Diagram.....	19
5.4 Perancangan Basis Data.....	20
5.5 Perancangan Algoritme	20
5.5.1 Perancangan TOPSIS	20
5.5.2 Perancangan Haversine.....	21
5.5.3 Perhitungan Manual	22
5.6 Perancangan Navigasi dan Antarmuka	26
5.6.1 Perancangan Screen Flow	26
5.6.2 Perancangan Screen Flow	27
BAB 6 IMPLEMENTASI	30
6.1 Spesifikasi Sistem	30
6.1.1 Spesifikasi Perangkat Keras.....	30
6.1.2 Spesifikasi Perangkat Lunak	31
6.2 Batasan-Batasan Implementasi	32
6.3 Implementasi Basis Data.....	32
6.4 Implementasi Class dan Assets pada file program	33
6.5 Implementasi Kode Program	33

6.5.1 Implementasi Kode Program Aplikasi Perangkat Bergerak	34
6.5.2 Implementasi Kode Algoritme Haversine	37
6.5.3 Implementasi Kode Program Algoritme TOPSIS	37
6.6 Implementasi Antarmuka	42
BAB 7 PENGUJIAN DAN ANALISIS	45
7.1 Pengujian Fungsional	45
7.2 Pengujian Non-Fungsional	46
7.2.1 Pengujian Validasi Algoritme	46
7.2.2 Pengujian Akurasi	47
BAB 8 KESIMPULAN DAN SARAN	52
8.1 Kesimpulan.....	52
8.2 Saran	53
DAFTAR REFERENSI	54
LAMPIRAN	56



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka Penelitian Sebelumnya	4
Tabel 4.1 Identifikasi Aktor	12
Tabel 4.2 Spesifikasi Kebutuhan Fungsional	13
Tabel 4.3 Spesifikasi Use Case.....	13
Tabel 4.4 Use Case Scenario Mendapatkan Rekomendasi Wisata	14
Tabel 4.5 Spesifikasi Kebutuhan Non Fungsional	15
Tabel 5.1 Data Lokasi	22
Tabel 5.2 Bobot Kriteria	22
Tabel 5.3 Sampel Data Alternatif	22
Tabel 5.4 Hasil Jumlah Alternatif Tiap Kriteria.....	23
Tabel 5.5 Hasil Normalisasi Matriks Alternatif.....	23
Tabel 5.6 Hasil Matriks Normalisasi Terbobot.....	24
Tabel 5.7 Hasil Solusi Ideal positif dan Solusi Ideal Negatif.....	24
Tabel 5.8 Hasil Perhitungan Separasi.....	25
Tabel 5.9 Hasil Perhitungan Kedekatan Relatif.....	25
Tabel 5.10 Hasil Perangkingan	26
Tabel 6.1 Spesifikasi Hardware Komputer.....	31
Tabel 6.2 Spesifikasi Mobile.....	31
Tabel 6.3 Spesifikasi Software Komputer	31
Tabel 6.4 Spesifikasi Software Mobile	31
Tabel 6.5 Struktur Tabel Basis Data	32
Tabel 6.6 Berkas Kelas dan Assets Layout.....	33
Tabel 7.1 Kasus Uji Fungsional mendapatkan Rekomendasi Wisata.....	45
Tabel 7.2 Hasil Pengujian Fungsional.....	46
Tabel 7.3 Data Alternatif.....	46
Tabel 7.4 Perpadanan Rekomendasi Aplikasi dan Manual	47
Tabel 7.5 Hasil Kecocokan Validasi Algoritme	47
Tabel 7.6 Hasil Pengujian Akurasi	48
Tabel 7.7 Hasil Kecocokan Pengujian Akurasi	49

DAFTAR GAMBAR

Gambar 3.1 Waterfall Model	10
Gambar 4.1 Diagram Analisis Kebutuhan	11
Gambar 4.2 Gambaran Aplikasi	12
Gambar 4.3 Use Case Diagram Kebutuhan Fungsional.....	13
Gambar 5.1 Diagram Perancangan Perangkat Lunak	16
Gambar 5.2 Activity Diagram Mendapatkan Rekomendasi Wisata.....	17
Gambar 5.3 Sequence Diagram Mendapatkan Rekomendasi Wisata.....	18
Gambar 5.4 Class Diagram Mendapatkan Rekomendasi Wisata.....	19
Gambar 5.5 Rancangan Basis Data	20
Gambar 5.6 Flowchart Algoritme TOPSIS	21
Gambar 5.7 Flowchart Algoritme Haversine.....	21
Gambar 5.8 Screenflow Aplikasi	26
Gambar 5.9 Antarmuka Menu Utama	27
Gambar 5.10 Antarmuka Bobot Prioritas	27
Gambar 5.11 Antarmuka Jenis Wisata	28
Gambar 5.12 Antarmuka Cari Wisata	29
Gambar 5.13 Antarmuka Rekomendasi	29
Gambar 6.1 Diagram Implementasi	30
Gambar 6.2 Hasil Implementasi Menu Utama	42
Gambar 6.3 Hasil Implementasi Bobot Prioritas.....	43
Gambar 6.4 Hasil Implementasi Jenis Wisata	43
Gambar 6.5 Hasil Implementasi Cari Wisata	44
Gambar 6.6 Hasil Implementasi Rekomendasi	44
Gambar 7.1 Diagram Pengujian	45

DAFTAR LAMPIRAN

LAMPIRAN A	56
A.1 Data Yang Digunakan.....	56
A.2 Data User	59



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kota Malang ialah kota kedua terbesar di Jawa Timur yang dikenal mempunyai julukan kota pariwisata. Kota Malang mempunyai beragam pilihan objek wisata. Tercatat banyak wisatawan domestik di Kota Malang tahun 2016 sebanyak 3.987.074 orang, dan wisatawan mancanegara sebanyak 9.535 orang (Sasongko, 2017). Namun, banyaknya wisata di Kota Malang menyebabkan banyak wisatawan bingung sehingga tidak dapat menentukan wisata yang ingin dikunjungi karena keterbatasan waktu dan dana. Untuk mengatasi permasalahan tersebut, diperlukan sebuah aplikasi pendukung keputusan untuk merekomendasi wisata berbasis *mobile* sehingga wisatawan diharapkan mendapat rekomendasi dengan kriteria yang diinginkan darimana saja dan kapan saja.

Sistem Pendukung Keputusan (SPK) ialah sebuah sistem yang memiliki tujuan dalam menunjang pengguna dalam mendukung pengambilan keputusan suatu masalah, hingga meningkatkan efektifitas dalam pengambilan keputusan (Indrawasti, Barizana, & Prottasof, 2017). Pada penelitian ini sistem pendukung keputusan dipakai guna merekomendasikan wisata dengan kriteria kriteria yaitu Jarak, Harga, Rating dan Jumlah Review.

Sistem rekomendasi wisata merupakan permasalahan yang memiliki banyak kriteria (multikriteria). Salah satu algoritme yang bisa menyelesaikan permasalahan multikriteria adalah algoritme TOPSIS (*Technique for Order Performance by Similarity to Ideal Solution*) (Kurniasih, 2013). TOPSIS memakai landasan bahwa alternatif yang dipilih wajib memiliki jarak paling dekat terhadap solusi ideal positif dan paling jauh terhadap solusi ideal negatif dengan menggunakan jarak Euclidean dalam mencari kedekatan relatif dari suatu alternatif dengan solusi yang optimal (Sudarsono dkk, 2016). Selain itu, dalam algoritme TOPSIS memiliki kompleksitas algoritme yang lebih rendah dibandingkan dengan algoritme lain, sehingga cocok digunakan untuk platform *mobile* yang memiliki keterbatasan daya dan komputasi. Hal ini dikarenakan memiliki konsep yang simpel, dan juga perhitungan yang efisien (Octavia, Widodo, Kasih, 2017). Karena penggunaan metode TOPSIS dalam *smartphone* tidak membutuhkan resource yang berat, sehingga kebutuhan hardware dalam metode ini juga rendah dan memungkinkan untuk diterapkan dalam *smartphone* setiap orang (Kozina, Volkova, Horpenko, 2018).

Berdasarkan permasalahan yang dijelaskan sebelumnya, peneliti berniat membangun sebuah sistem rekomendasi aplikasi wisata dalam kota malang dengan basis *mobile* dengan menggunakan algoritme TOPSIS. Sehingga diharapkan nantinya dapat membantu wisatawan yang sedang berkunjung ke Malang menentukan tujuan wisata seperti yang diinginkan.

1.2 Rumusan Masalah

1. Bagaimana perancangan aplikasi rekomendasi wisata dalam kota Malang berbasis lokasi?
2. Bagaimana implementasi aplikasi rekomendasi wisata dalam kota Malang?
3. Bagaimana pengujian aplikasi rekomendasi wisata dalam kota Malang?

1.3 Tujuan

Tujuan yang diharapkan bakal diraih atas dilakukannya penelitian ini, yaitu:

1. Mengembangkan aplikasi berbasis android yang mengimplementasikan algoritme TOPSIS guna pencarian rekomendasi wisata di dalam kota Malang sesuai dengan keinginan dan kriteria pengguna.

1.4 Manfaat

1. Memudahkan wisatawan yang sedang mencari tempat wisata di dalam kota Malang yang sesuai dengan kriteria dan keinginan setiap saat memakai aplikasi Android.
2. Membantu wisatawan dengan memberikan rekomendasi berupa daftar rekomendasi tempat wisata di dalam kota Malang disertai dengan detail tempat wisata hingga melakukan navigasi menuju tempat wisata yang dipilih.

1.5 Batasan Masalah

1. Data objek wisata yang dipakai berada di dalam Kota Malang saja.
2. Penggunaan jenis kriteria yang dipakai dalam perhitungan rekomendasi wisata adalah jarak wisata, harga tiket wisata, rating wisata dan jumlah review wisata.
3. Algoritme untuk merekomendasikan yang dipakai ialah algoritme TOPSIS.
4. Aplikasi *webservice* sudah dibuat dan dapat langsung digunakan untuk mendukung penelitian ini. Tanpa perlu pembahasan perancangan dan implementasi *webservice*.
5. Status koneksi internet dan GPS pada *smartphone* dianggap kondisi tersambung dan menyala sehingga bisa dipakai oleh aplikasi secara langsung.

1.6 Sistematika Pembahasan

BAB I PENDAHULUAN

Bab ini mendeskripsikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat dari penelitian, hingga sistematika penulisan mengenai pentingnya penelitian “Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi”.

BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai kajian pustaka penelitian sebelumnya juga dasar teori yang akan digunakan dalam penelitian “Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi”.

BAB III METODOLOGI PENELITIAN

Bab ini menjabarkan tentang tahap tahap dan metode yang bakal dikerjakan di penelitian, antara lain : Strategi Penelitian, Subjek Peneletian, Lokasi Penelitian Metode Pengumpulan Data, Peralatan Pendukung, hingga Metode Pengembangan Sistem

BAB IV ANALISIS DAN PERANCANGAN

Bab ini membahas tentang bagaimana perancangan dan analisis dari aplikasi dalam penelitian “Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi”.

BAB V IMPLEMENTASI

Bab ini menjabarkan mengenai penerapan dari aplikasi dalam penelitian “Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi” yang telah dirancang sebelumnya..

BAB VI PENGUJIAN

Bab ini menjelaskan berkenaan teknik yang dipakai dalam pengujian aplikasi. Pengujian yang akan dipakai yaitu pengujian fungsional, validasi algoritme, hingga akurasi.

BAB VII PENUTUP

Bab ini menjabarkan tentang kesimpulan dan saran dari penelitian “Implementasi TOPSIS pada Sistem Rekomendasi Tempat Wisata Dalam Kota Malang Berbasis Lokasi”.

BAB 2 LANDASAN KEPUSTAKAAN

Dalam bagian ini menjabarkan berkenaan kajian pustaka penelitian yang telah ada lebih dahulu dan juga dasar teori yang nantinya dipakai sebagai pedoman dalam penelitian.

2.1 Kajian Pustaka

Terdapat penelitian terdahulu yang memakai algoritme TOPSIS dalam menyelesaikan permasalahannya. Dalam Tabel 2.1 menerangkan mengenai penelitian yang telah ada lebih dahulu dan menggunakan algoritme TOPSIS.

Tabel 2.1 Kajian Pustaka Penelitian Sebelumnya

No	Peneliti	Judul Penelitian	Hasil
1	Yuliya Kozina, Natalya Volkova, Daniil Horpenko	Mobile Application for Decision Support in Multi-Criteria Problems	Penggunaan metode TOPSIS dalam smartphone adalah layak, karena perhitungan dalam metode tersebut tidak membutuhkan resource yang berat, sehingga kebutuhan hardware dalam metode ini juga rendah dan memungkinkan untuk diterapkan dalam smartphone setiap orang
2	Avriana Indarwasti, Barizana S.A, Prottasof G.K	Sistem Pendukung Keputusan Pemilihan Tempat Kuliner di Depok dengan TOPSIS	Aplikasi berhasil memberikan rekomendasi merekomendasikan tempat makan / kuliner di Depok memakai algoritme TOPSIS dengan Kriteria yang dipakai di perhitungan rekomendasi ialah Rasa, Variasi Menu, Waktu Operasional, Area, Fasilitas.
3	Dhanie Octavia, Dr. Suryo Widodo.,M.Pd, Patmi Kasih, M.Kom	Aplikasi Pencarian Tempat Futsal dengan Menggunakan Metode TOPSIS	Aplikasi berhasil memunculkan hasil pencarian lapangan futsal dengan memakai algoritme TOPSIS berdasarkan perankingan dengan kriteria Nilai lapangan, harga, waktu, hingga jenis lapangan.

Aplikasi yang akan dikembangkan dalam penelitian ini menggunakan metode TOPSIS yang diharapkan merekomendasikan wisata di dalam kota Malang berdasarkan parameter kriteria yang dipilih oleh wisatawan. Lebih dari itu, aplikasi ini juga dapat menampilkan informasi wisata meliputi jarak, harga tiket masuk dan rute menuju wisata secara lengkap. Kriteria yang akan digunakan dalam mencari

rekomendasi adalah jarak, harga tiket, rating dan jumlah review pada objek wisata.

2.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) ialah sebuah sistem yang dibuat dan memiliki tujuan dalam menunjang pengguna dalam mendukung pengambilan keputusan suatu masalah, hingga meningkatkan efektifitas dalam pengambilan keputusan. SPK biasanya digunakan oleh manajer sebuah perusahaan dalam membantu memilih sebuah keputusan (Indrawasti, Barizana, & Prottasof, 2017).

2.3 Technique For Order Preference by Similiarity (TOPSIS)

TOPSIS ialah algortime yang dikenalkan oleh Yoon dan Hwang dimana pada algoritme ini memilih alternatif berjarak paling dekat terhadap solusi ideal positif dan juga berjarak dengan solusi ideal negatif paling jauh.

Langkah langkah perhitungan TOPSIS :

1. Menentukan Matriks Normalisasi

Matriks normalisasi dihitung memakai jarak Euclidean seperti dalam Persamaan (2.1).

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

dengan:

r_{ij} = hasil normalisasi matriks R;

$i = 1, 2, 3, \dots, m$;

$j = 1, 2, 3, \dots, m$;

2. Membuat Matriks Normalisasi Terbobot

Mengerjakan pembobotan matriks yang sudah dinormalisasikan memakai bobot $W = (w_1, w_2, \dots, w_n)$ seperti dijabarkan dalam Persamaan (2.2).

$$V = \begin{bmatrix} w_1 r_{11} & \cdots & w_n r_{1n} \\ \vdots & \ddots & \vdots \\ w_1 r_{m1} & \cdots & w_n r_{mn} \end{bmatrix} \quad (2.2)$$

dimana:

$i = 1, 2, 3, \dots, m$;

$j = 1, 2, 3, \dots, n$;

3. Mencari solusi ideal positif dan solusi ideal negatif.

Dalam menentukan Solusi ideal positif juga negatif, sebelumnya ditentukan dulu apakah bersifat keuntungan / biaya. Seperti dalam Persamaan (2.3) dan Persamaan (2.4).

$$A^+ = \{(\max v_{ij} | j \in J) \quad (\min v_{ij} | j \in J'), i = 1, 2, 3, \dots, m\} = \{v_1^+, v_2^+, \dots, v_m^+\} \quad (2.3)$$

$$A^- = \{(\max v_{ij} | j \in J) \quad (\min v_{ij} | j \in J'), i = 1, 2, 3, \dots, m\} = \{v_1^-, v_2^-, \dots, v_m^-\} \quad (2.4)$$

dimana:

v_{ij} = elemen matriks V baris ke-i dan kolom ke-j;

$J = \{j = 1, 2, 3, \dots, m \text{ dan } j \text{ kriteria bersifat keuntungan}\};$

$J' = \{j = 1, 2, 3, \dots, m \text{ dan } j \text{ kriteria bersifat biaya }\};$

4. Menghitung separasi

Menghitung separasi dilakukan dengan mencari jarak setiap nilai terhadap matriks solusi ideal positif dan negatif. Nilai alternatif ke I dengan solusi ideal positif memiliki jarak yang dihitung dengan Persamaan (2.5) Lalu, solusi ideal negatif diperoleh memakai Persamaan (2.6).

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, \text{ dengan } i = 1, 2, 3, \dots, n; \quad (2.5)$$

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \text{ dengan } i = 1, 2, 3, \dots, n; \quad (2.6)$$

5. Mengukur kedekatan relatif atas solusi ideal.

Untuk menghitung kedekatan relatif dari alternatif A^+ dengan solusi ideal A^- digunakan Persamaan (2.7)

$$C_i^+ = \frac{S_i^-}{S_i^- + S_i^+}, \text{ dengan } 0 < C_i^+ < 1 \text{ dan } i = 1, 2, 3, \dots, m; \quad (2.7)$$

6. Merangking alternatif

Setelah melakukan perhitungan kedekatan alternatif terhadap solusi ideal, didapatkan nilai C_i^+ . Semakin besar nilai C_i^+ , maka solusi semakin baik. Merangking alternatif dilakukan dengan mengurutkan nilai C_i^+ pada setiap alternatif dari yang paling besar (Purnomo, Sihwi, & Anggrainingsih, 2013).

2.4 Haversine Formula

Algoritme Haversine ialah sebuah persamaan yang dipakai dalam navigasi dimana formula ini dapat memperoleh jarak antara dua titik dalam permukaan bola (bumi) dengan dasar bujur (longitude) dan lintang (latitude) (Setiawan, H., 2016).

Persamaan haversine ditunjukkan dalam Persamaan (2.8) berikut :

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{lat_2 - lat_1}{2} \right) + \cos(lat_1) \cos(lat_2) \sin^2 \left(\frac{long_2 - long_1}{2} \right)} \right) \quad (2.8)$$

Dimana :

d = jarak;

Lat1 = Latitude Awal;

Lat2 = Latitude Akhir / Tujuan;

Long1 = Longitude Awal;

Long2 = Longitude Akhir / Tujuan;

R = Radius Bumi yang dipakai sebesar 6371 Km;

2.5 Global Positioning System (GPS)

Global Positioning System (GPS) ialah perangkat yang memiliki fungsi guna mendapatkan posisi secara tepat koordinat bumi yang menerima langsung sinyal dari satelit. Saat ini perangkat GPS menggunakan peta sehingga dapat dimanfaatkan sebagai navigasi di darat, laut dan udara. GPS memiliki fungsi bermacam macam mulai dari kebutuhan Militer seperti mengetahui posisi pasukan atau menutun arah bom, Sebagai navigasi dalam berkendara, Sistem Informasi Geografis, Sistem Pelacakan Kendaraan, hingga sebagai pemantau gempa (Purnama, 2011).

2.6 Android

Android ialah *Operating System* dalam perangkat *mobile* yang memiliki basis Linux. Android bersifat *Open Source* untuk *developer* dalam membuat aplikasi. Sebagian besar produsen *smartphone* saat ini telah memproduksi *smartphone* dengan sistem operasi Android karena sistem operasinya yang bersifat *open source* sehingga dapat didistribusikan secara bebas dan dipakai (Setiawan, Andjarwirawan, & Handojo, 2013).

2.7 Android Studio

Android Studio merupakan IDE dalam melakukan pengembangan aplikasi berbasis android resmi. Android Studio memiliki basis IntelliJ IDEA yang digunakan dalam mengembangkan berbagai perangkat lunak komputer (Satyaputra, 2016).

2.8 JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) ialah sebuah struktur saat melakukan perpindahan data dengan sifat ringkas, gampang dibaca, gampang ditulis, gampang dibuat, hingga mudah di terjemahkan. JSON tidak bergantung kepada bahasa pemrograman yang digunakan dikarenakan gaya bahasa yang digunakan umum. JSON terdiri dari dua bentuk, bentuk pertama Kumpulan pasangan nama/nilai sedangkan bentuk yang kedua merupakan Daftar nilai yang terurutkan. Struktur data ini merupakan struktur data universal didukung semua bahasa pemrograman moderen (json.org).

2.9 Web Service

Web service ialah sebuah *software* berbasis web dan dikembangkan dalam mendukung proses interaksi antar mesin melalui internet / jaringan. *Web Service* mempunyai fasilitas guna integrasi data kollaborasi informasi yang dimungkinkan untuk diakses oleh berbagai pihak melaui internet menggunakan berbagai macam teknologi yang digunakan oleh pengguna (Sutanta, 2012).



BAB 3 METODOLOGI

Pada bagian metodologi akan menjabarkan tentang tahap dan tata cara yang akan dikerjakan, yaitu: strategi penelitian, subjek penelitian, lokasi penelitian, metode pengumpulan data, peralatan pendukung, hingga metode pengembangan.

3.1 Strategi Penelitian

Strategi penelitian akan dikerjakan dengan studi literatur guna mempelajari teori yang digunakan pada penulisan skripsi dengan sumber dari jurnal, buku, dan *paper* terdahulu. Pada penelitian ini, topik yang dipelajari, yaitu:

1. Kajian pustaka dari penelitian terdahulu
2. Sistem Pendukung Keputusan
3. *Technique for Order Preference by Similiarity* (TOPSIS)
4. *Haversine Formula*
5. *Global Positioning System* (GPS)
5. Android
6. Android Studio
7. *Javascript Object Notation* (JSON)
8. Web Service

3.2 Subjek Penelitian

Subjek yang akan dipakai dalam penelitian ini merupakan tempat wisata yang berada dalam kota Malang. Data yang diambil adalah koordinat lokasi wisata, harga masuk wisata, jumlah review, dan nilai rating dari *Google*.

3.3 Lokasi Penelitian

Lokasi dalam penelitian ini terletak dalam Kota Malang saja, termasuk data yang diperlukan hanya terletak di daerah Kota Malang.

3.4 Metode Pengumpulan Data

Data objek wisata dikumpulkan melalui observasi langsung pada beberapa tempat wisata dan informasi dari *Google maps*. Pada pengujian akurasi, dikumpulkan 30 data yang dikerjakan melalui wawancara langsung terhadap 30 pakar berbeda di berbagai lokasi di Kota Malang pada 14 Januari 2019 hingga 22 Februari 2019.

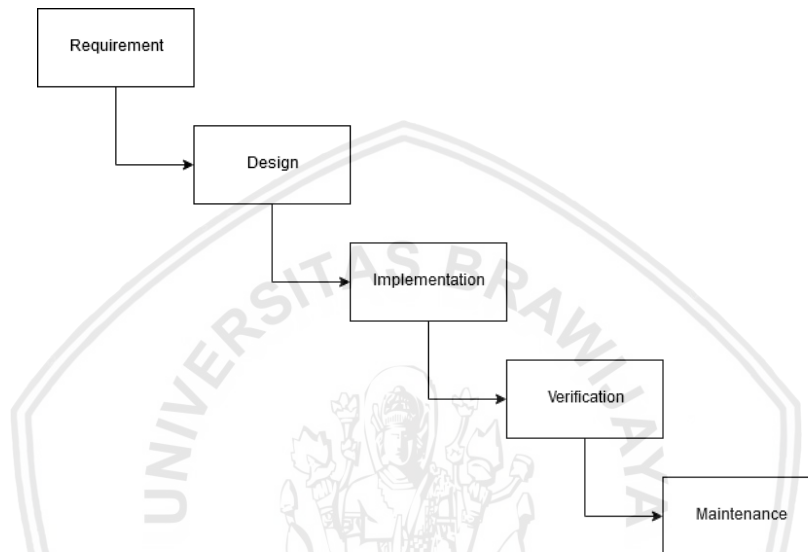
3.5 Peralatan Pendukung

Peralatan pendukung yang akan dipakai pada penelitian kali ini dalam membangun sistem rekomendasi wisata dalam kota Malang terdiri dari peralatan *hardware* dan *Software*. Untuk *hardware*, yang digunakan adalah *Personal*

Computer (PC) dan *smartphone*. Kemudian *software* yang digunakan ialah IDE Android Studio.

3.6 Metode Pengembangan Sistem

Metode pengembangan sistem dilakukan pada penelitian ini yang dipakai untuk membangun sistem adalah menggunakan metode *waterfall*. Metode ini dipilih karena kebutuhan yang ingin dituju telah dapat dijabarkan dan tidak berubah. Gambar 3.1 berikut merupakan langkah metode *waterfall* :

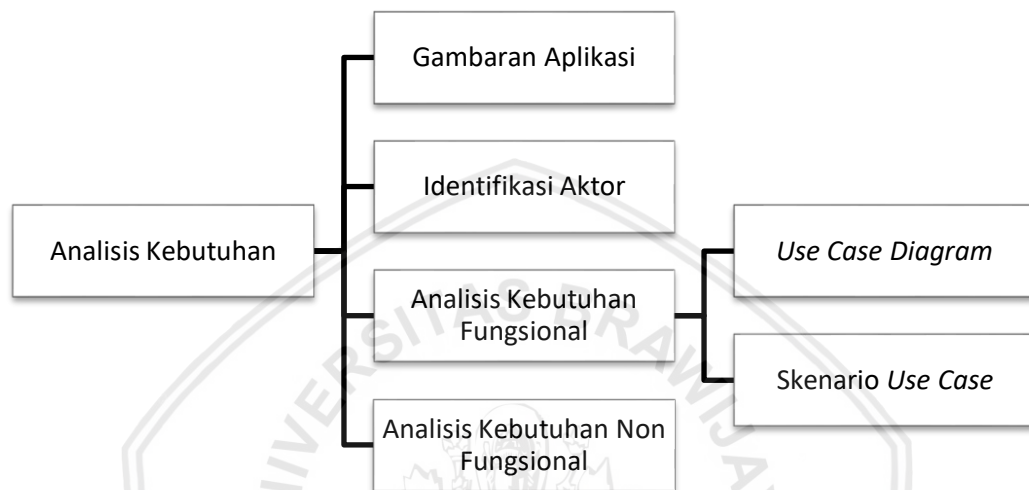


Gambar 3.1 Waterfall Model

Dalam penelitian ini, Tahap *Requirement* dilakukan dengan melakukan analisis dari kebutuhan apa saja yang harus dipenuhi oleh sistem. Analisis dilakukan dengan membuat gambaran aplikasi, identifikasi aktor, analisis kebutuhan fungsional, dan analisis kebutuhan non fungsional. Selanjutnya tahap *Design* dilakukan dengan membuat perancangan perangkat lunak berdasarkan analisis sebelumnya mulai dari *activity diagram*, *sequence diagram*, *class diagram*, hingga perancangan antarmuka. Tahap ketiga *Implementation* dilakukan dengan melakukan penerapan terhadap kode, kelas, dan basis data yang telah dirancang sebelumnya hingga menjadi sebuah satu aplikasi yang utuh dapat berjalan dengan baik. Selanjutnya tahap *verification* dilakukan dengan melakukan pengujian untuk mengecek apakah sistem yang dibangun telah diimplementasikan sesuai dengan perancangan sebelumnya. Pengujian yang dilakukan dalam penelitian ini merupakan pengujian fungsional yaitu *blackbox testing* untuk menguji kebutuhan fungsional sistem telah diterapkan dengan benar atau tidak dan untuk kebutuhan non fungsional menggunakan pengujian validasi algoritma dan pengujian akurasi. Terakhir, tahap *Maintenance* merupakan tahap pemeliharaan sistem dan memperbaiki *bug* yang tidak ditemukan dalam tahap pengujian. Namun dalam penelitian ini tidak dilakukan tahap *maintenance* karena berakhir dalam tahap *verification*.

BAB 4 ANALISIS KEBUTUHAN

Dalam bagian ini akan menjabarkan mengenai analisis dalam aplikasi rekomendasi wisata dalam kota Malang. Tahapannya akan dijabarkan dalam Gambar 4.1 berikut.

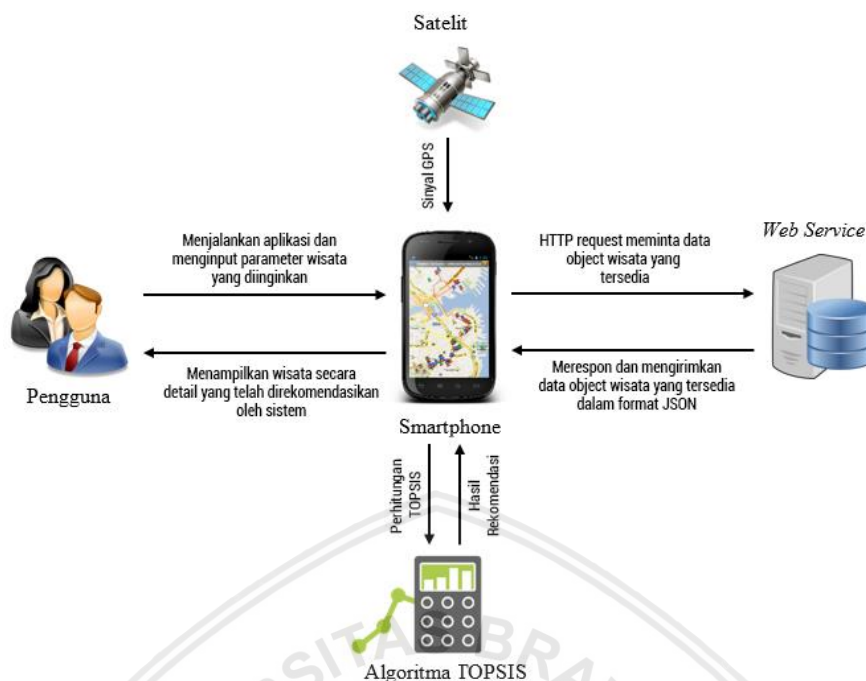


Gambar 4.1 Diagram Analisis Kebutuhan

Dalam langkah ini, dibuat indentifikasi keperluan aplikasi dengan sejumlah langkah analisis, dengan tujuan demi mengetahui tentang apa saja yang wajib tersedia di dalam sistem agar mencukupi keperluan *user*.

4.1 Gambaran Aplikasi

Gambaran aplikasi adalah gambar secara luas mengenai sistem yang hendak dikembangkan dan merupakan langkah pertama dalam merancang sistem. Gambaran aplikasi rekomendasi wisata dalam kota Malang ditunjukkan pada Gambar 4.2.



Gambar 4.2 Gambaran Aplikasi

User membuka aplikasi di *smartphone* yang dimiliki. Aplikasi memerlukan internet juga GPS agar menyala sehingga aplikasi bisa mengambil data dari *web service* dan mengetahui lokasi pengguna saat ini. Setelah membuka aplikasi, sistem akan memunculkan menu cari wisata yang terdapat fitur dimana *user* dapat memasukkan input parameter yang diinginkan oleh pengguna. Parameter yang dipakai aplikasi ini ialah harga, jarak, rating dan jumlah review.

Data parameter yang dimasukkan user yang berupa harga, jarak, rating dan jumlah *review* yang diinginkan pengguna akan digabungkan dengan koordinat lokasi pengguna saat ini untuk dilakukan perhitungan memakai algoritme TOPSIS guna memperoleh rekomendasi yang cocok melalui parameter yang dimasukkan berdasarkan data wisata yang telah didapatkan dari *web service* dalam format JSON. Setelah itu hasil perhitungan akan direkomendasikan dalam bentuk detail wisata secara lengkap.

4.2 Identifikasi Aktor

Identifikasi aktor akan dikerjakan melalui pengenalan kepada aktor yang berhubungan kepada sistem seperti terdapat dalam Tabel 4.1 berikut.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna Aplikasi	<i>User</i> bisa melakukan input kriteria berupa harga, jarak, rating dan jumlah review. Pengguna kemudian memperoleh hasil rekomendasi wisata yang diinginkan.

4.3 Analisis Kebutuhan Fungsional

Dalam langkah ini menjabarkan mengenai kebutuhan fungsional seperti dalam Tabel 4.2 berikut.

Tabel 4.2 Spesifikasi Kebutuhan Fungsional

Identifier	Kebutuhan Fungsional	Nama Use Case
SKF-01	Aplikasi perlu menyediakan fitur untuk merekomendasikan objek wisata.	Mendapatkan Rekomendasi

4.3.1 Use Case Diagram

Usecase diagram ialah sebuah gambaran berbentuk diagram dan dibangun searah dengan kebutuhan fungsional pada tahap sebelumnya dan digunakan untuk menjelaskan tentang kebutuhan yang bisa dikerjakan aktor dalam sistem. Gambar 4.3 menjabarkan mengenai *usecase diagram* yang dipakai pada penelitian ini.



Gambar 4.3 Use Case Diagram Kebutuhan Fungsional

4.3.2 Spesifikasi Use Case

Spesifikasi *use case* menjabarkan rincian *use case* mendapatkan rekomendasi (SKF-01) dalam *usecase diagram* seperti dijelaskan dalam Tabel 4.3 berikut.

Tabel 4.3 Spesifikasi Use Case

No.	Spesifikasi Use case	Deskripsi
1.	Mendapatkan rekomendasi wisata.	Fitur memunculkan rekomendasi wisata memakai kriteria yang diinputkan oleh user.

4.3.3 Skenario Use Case

Use Case Scenario menjabarkan mengenai rincian spesifikasi *use case* pada Tabel 4.3. Dalam *usecase scenario* ini menjabarkan mulai dari aktor yang terlibat,

tujuan, deskripsi, *pre-condition*, *post-condition*, hingga aliran usecase. *Usecase Scenario* Mendapatkan Rekomendasi Wisata dijabarkan dalam Tabel 4.4 berikut.

Tabel 4.4 Use Case Scenario Mendapatkan Rekomendasi Wisata

Nama	Mendapatkan Rekomendasi Wisata
Aktor	Pengguna Aplikasi
Tujuan	Guna mendapatkan rekomendasi objek wisata berdasarkan kriteria yang diinputkan user (jarak, harga, rating dan jumlah review)
Deskripsi	Menjelaskan bagaimana pengguna melakukan <i>input</i> kriteria pencarian, menghitung, hingga menampilkan hasil rekomendasi objek wisata.
Pre-condition	Aktor telah menjalankan aplikasi. Kemudian memilih menu cari wisata.
Post-condition	Sistem akan menampilkan rekomendasi wisata berdasarkan hasil perhitungan memakai kriteria yang dimasukkan oleh user.
Aliran Utama	
Aksi Aktor	Respon Sistem
1. Menjalankan aplikasi.	2. (a) Ada internet dan GPS hidup. Aplikasi mengunduh data objek wisata dari webservice
	3. Membuka menu utama
4. Memilih menu cari wisata.	5. Menampilkan halaman form menentukan prioritas kriteria.
6. Mengisi form prioritas kriteria, kemudian tekan selanjutnya pada halaman form.	7. (a) form prioritas kriteria telah diisi dengan benar. Menampilkan halaman form menentukan jenis wisata yang diinginkan
8. Mengisi form jenis wisata, kemudian tekan cari wisata pada halaman form.	9. (a) form jenis wisata telah diisi dengan benar. Mulai melakukan proses mendapatkan rekomendasi wisata.
	10. (a) Ada internet dan GPS hidup. Mendapatkan koordinat lokasi pengguna dan melakukan perhitungan TOPSIS berdasarkan kriteria yang diinputkan user.
	11. Menampilkan hasil rekomendasi wisata secara detail.
Aliran Alternatif	
Aksi Aktor	Respon Sistem

	2. (b) Tidak ada internet dan GPS mati. Aplikasi tidak dapat mengunduh data objek wisata dan aplikasi berhenti
	7. (b) Form kriteria tidak diisi dengan benar / ada yang kosong. Menampilkan pesan peringatan.
	9. (b) Form jenis wisata tidak diisi dengan benar / ada yang kosong. Menampilkan pesan peringatan.
	10. (b) Tidak ada internet dan GPS mati. Perhitungan gagal dilakukan. Sehingga tidak dapat memunculkan rekomendasi wisata

4.4 Analisis Kebutuhan Non Fungsional

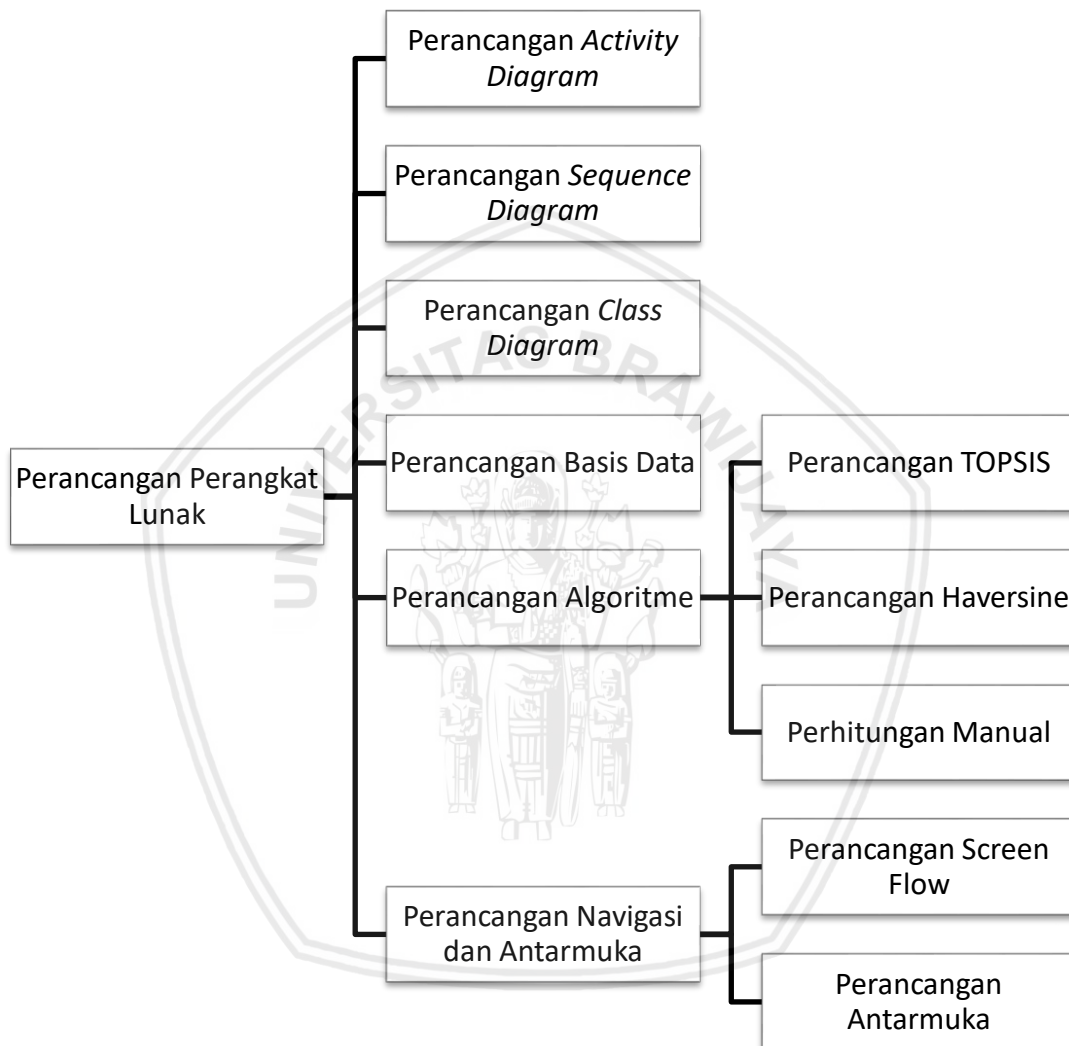
Bagian ini menjelaskan mengenai analisis kebutuhan yang tak terdapat pada kebutuhan, tetapi diperlukan dalam sistem seperti dijabarkan dalam Tabel 4.5 berikut.

Tabel 4.5 Spesifikasi Kebutuhan Non Fungsional

Kode	Nama	Deskripsi
SKNF-001	Validasi Algoritme	Mencocokkan antara rekomendasi sistem dengan rekomendasi perhitungan manual. Metode TOPSIS dianggap berhasil diterapkan apabila hasilnya sama.
SKNF-002	Akurasi	Melakukan perbandingan antara hasil rekomendasi sistem terhadap keinginan pengguna. Hasil akan dianggap akurat apabila hasil yang direkomendasikan sistem sesuai dengan keinginan pengguna.

BAB 5 PERANCANGAN PERANGKAT LUNAK

Dalam bagian ini menjabarkan mengenai perancangan aplikasi perangkat lunak rekomendasi wisata dalam kota Malang. Tahapannya akan dijabarkan dalam Gambar 5.1 berikut.

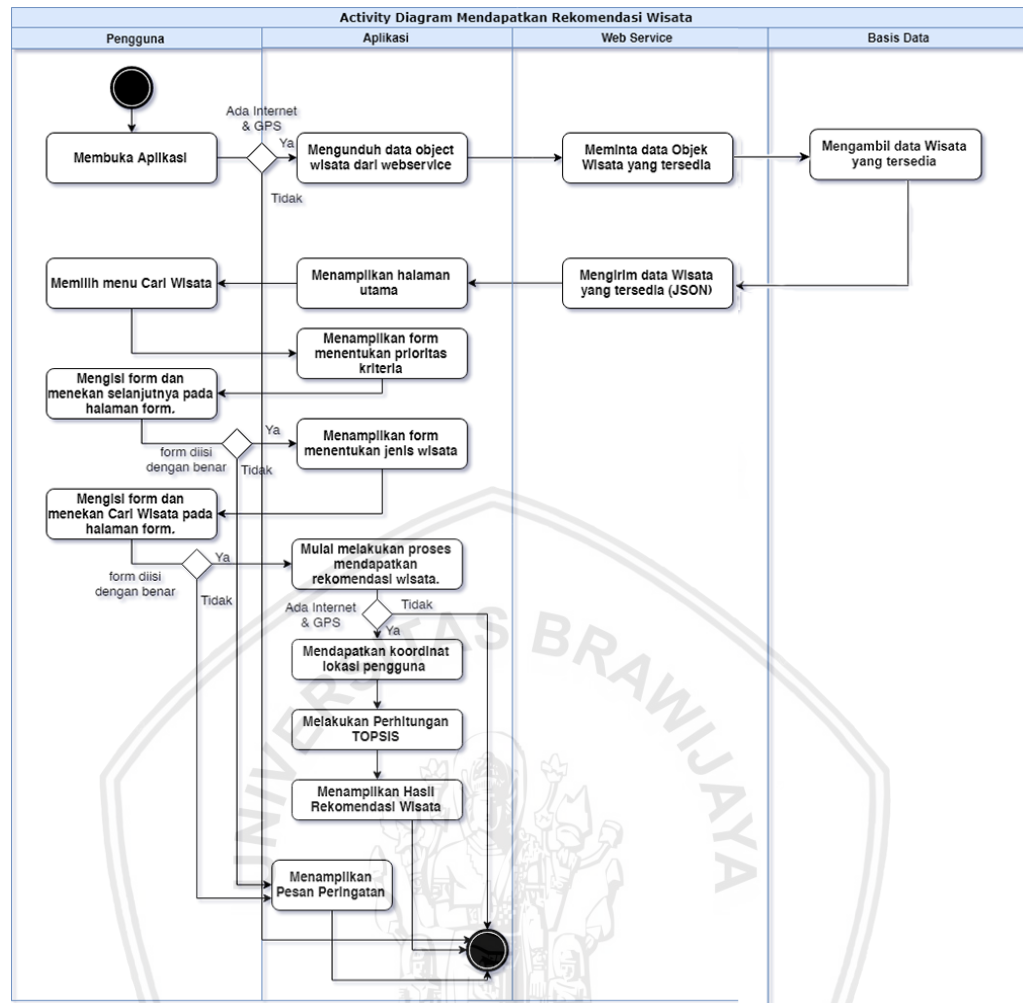


Gambar 5.1 Diagram Perancangan Perangkat Lunak

Dalam bagian ini mendeskripsikan tahapan perancangan aplikasi yang nantinya akan dibangun dimulai dari merancang *activity diagram*, *sequence diagram*, *class diagram*, basis data, algoritme, hingga antarmuka.

5.1 Perancangan *Activity Diagram*

Activity diagram adalah sebuah representasi model kegiatan yang dikerjakan *user* kepada sistem yang aktif berbasis *use case scenario* sebelumnya. Gambar 5.2 berikut merupakan *activity diagram* yang dipakai.



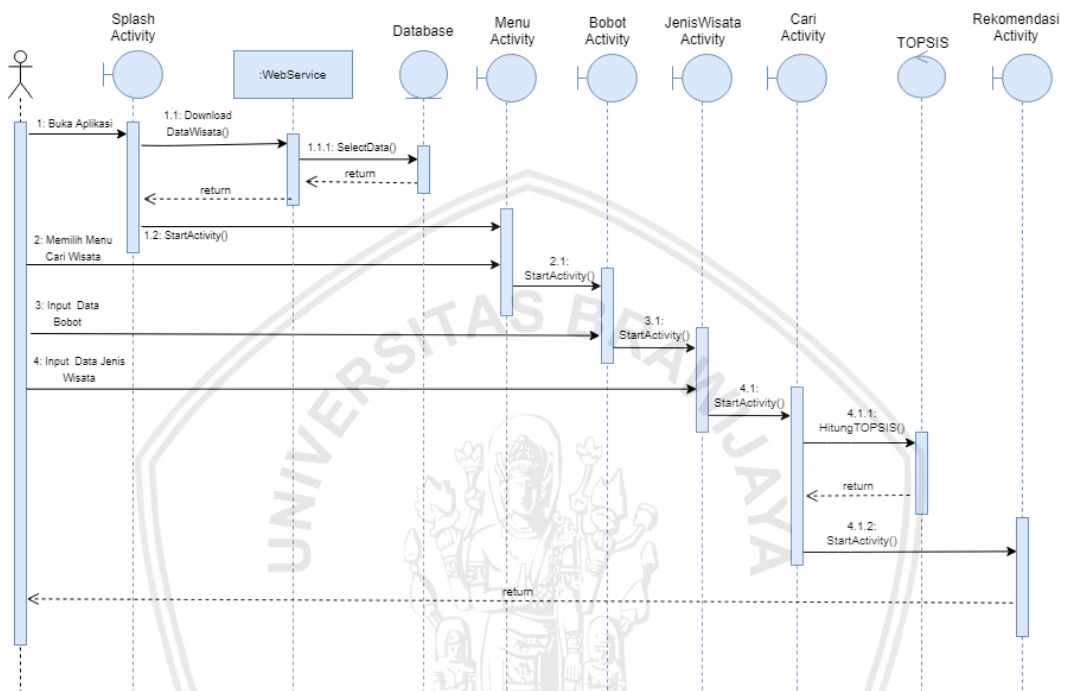
Gambar 5.2 Activity Diagram Mendapatkan Rekomendasi Wisata

Gambar 5.2 menjabarkan tentang kegiatan yang dikerjakan oleh *user* dan aplikasi. Pertama, *user* mengaktifkan aplikasi dahulu, setelah itu aplikasi kemudian memunculkan *splash screen* dan mengunduh data wisata dari *webservice*. Namun, apabila koneksi internet tidak tersambung, maka aplikasi akan keluar. Ketika mengunduh data berhasil, maka aplikasi akan menampilkan halaman utama yang terdapat menu Cari Wisata. Kemudian, Pengguna memilih menu Cari Wisata lalu sistem menampilkan halaman untuk menentukan prioritas bobot dari kriteria yang diinginkan oleh pengguna. Setelah pengguna mengisi isian bobot dan memencet tombol selanjutnya, aplikasi selanjutnya mengecek form telah diisi dengan tepat. Apabila belum diisi secara tepat, aplikasi kemudian memunculkan pesan peringatan, sebaliknya jika telah tepat aplikasi selanjutnya memunculkan halaman untuk menentukan jenis wisata apa yang diinginkan oleh pengguna. Setelah pengguna mengisi form dan menekan tombol cari wisata, sistem akan kembali memeriksa apakah form telah diisi dengan benar. Apabila belum diisi dengan benar, kemudian sistem akan memunculkan pesan peringatan, namun ketika telah diisi dengan tepat aplikasi kemudian memanggil halaman cari wisata dan akan memperoleh lokasi pengguna memakai GPS. Dari data GPS yang diperoleh, akan dilakukan perhitungan TOPSIS untuk mendapatkan rekomendasi. Setelah

perhitungan berhasil dilakukan, sistem akan menampilkan hasil pada halaman rekomendasi wisata.

5.2 Perancangan *Sequence Diagram*

Sequence diagram ialah gambaran yang menjabarkan tentang hubungan antar objek dan kelas pada sistem.

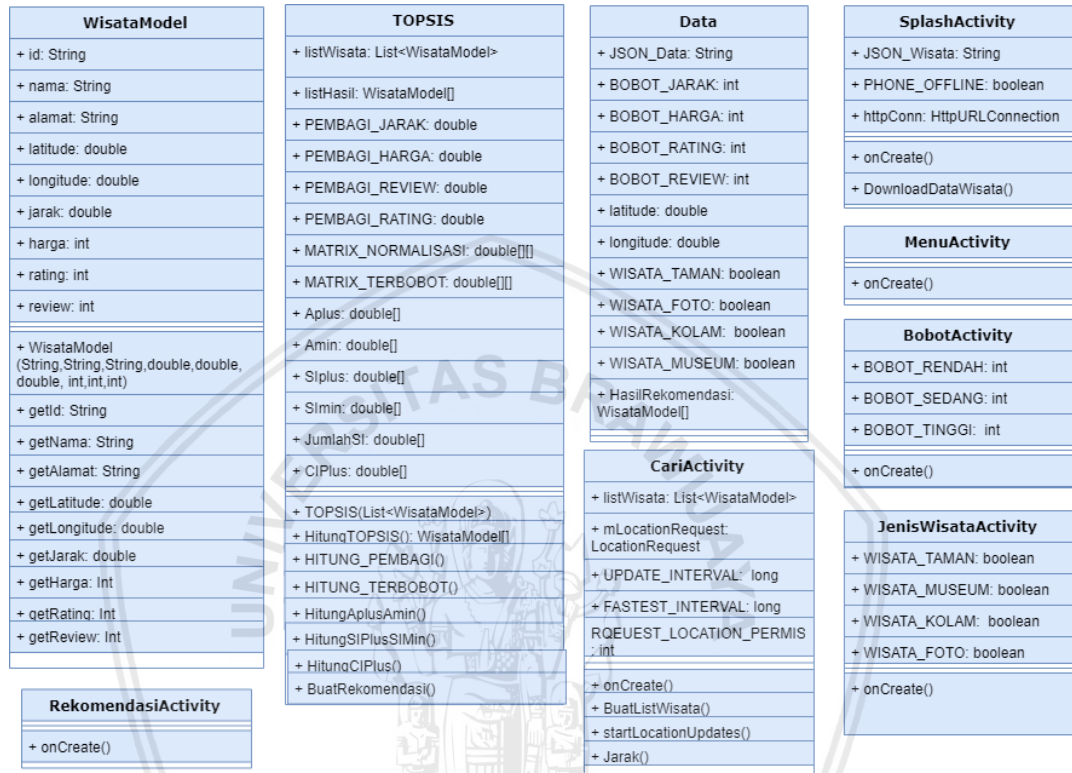


Gambar 5.3 *Sequence Diagram* Mendapatkan Rekomendasi Wisata

Gambar 5.3 menunjukkan *sequence diagram* untuk mendapatkan rekomendasi wisata dengan skenario usecase Mendapatkan Rekomendasi Wisata. Ketika *user* menjalankan aplikasi selanjutnya akan membuka kelas SplashActivity. Dimana pada SplashActivity akan melakukan pengunduhan data wisata dari *webservice* yang berupa JSON. Ketika pengunduhan selesai, sistem akan membuka MenuActivity. Ketika pengguna memilih menu Cari Wisata, aplikasi kemudian membuka kelas BobotActivity. Dan ketika *user* telah menginput bobot kriteria dan klik selanjutnya, maka sistem akan memanggil kelas JenisWisataActivity yang digunakan untuk mengisi jenis wisata yang diinginkan oleh pengguna. Setelah diisi dengan benar, maka sistem akan memanggil CariActivity untuk mulai melakukan perhitungan TOPSIS dengan memanggil kelas TOPSIS. Setelah perhitungan selesai, aplikasi kemudian memunculkan rekomendasi wisata di dalam RekomendasiActivity.

5.3 Perancangan Class Diagram

Class diagram ialah bagan yang dipakai guna mendeskripsikan *class* yang dipakai aplikasi secara lengkap dengan nama, atribut dan fungsi pada kelas tersebut. *Class Diagram* yang digunakan dijelaskan dalam Gambar 5.4. berikut



Gambar 5.4 Class Diagram Mendapatkan Rekomendasi Wisata

Gambar 5.4 menjabarkan mengenai kelas beserta atribut dan operationnya yang digunakan dalam pengembangan aplikasi Rekomendasi Wisata. Kelas kelas yang digunakan, yaitu: *SplashActivity*, *MenuActivity*, *BobotActivity*, *JenisWisataActivity*, *CariActivity*, *Data*, *TOPSIS*, *WisataModel*, hingga *RekomendasiActivity*.

5.4 Perancangan Basis Data

Bagian ini memiliki tujuan guna menjelaskan bentuk penyimpanan data obyek wisata berbentuk struktur bagan yang saling berhubungan di setiap entitasnya.

Wisata		
PK	ID (int)	
	nama	varchar(100)
	latitude	decimal(11,8)
	longitude	decimal(11,8)
	harga	int(11)
	rating	int(11)
	review	int(11)
	alamat	varchar(255)
	kategori	varchar(255)

Gambar 5.5 Rancangan Basis Data

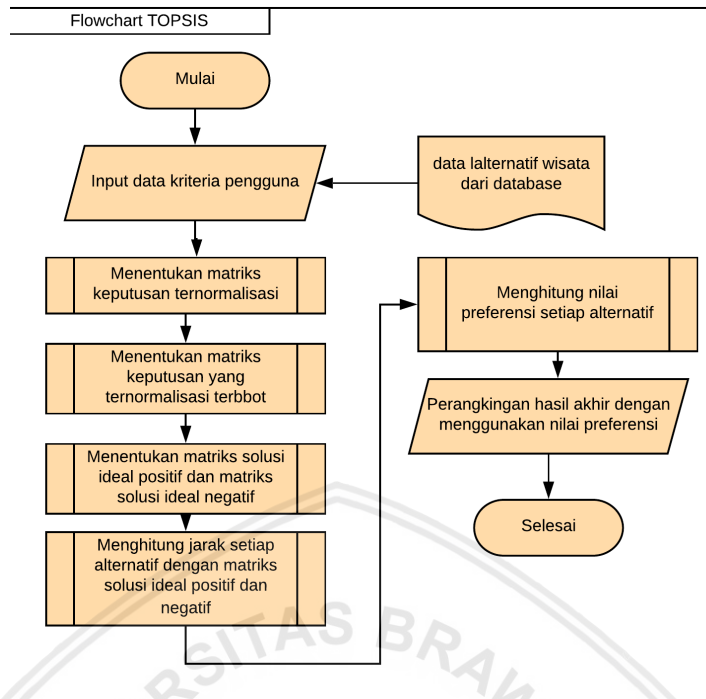
Gambar 5.5 menjabarkan tentang struktur *database* yang nantinya dipakai oleh aplikasi. Ada 1 tabel yaitu tabel wisata yang memiliki atribut id,nama, latitude, longitude, harga, rating, *review*, alamat dan kategori. Dimana nanti semua data obyek wisata akan disimpan dan dipanggil dari tabel tersebut oleh *webservice*.

5.5 Perancangan Algoritme

Bagian ini menjabarkan berkenaan langkah langkah algoritme yang dipakai pada sistem. Perancangan algoritme yang dipakai diantaranya perancangan algoritme TOPSIS dan perancangan algoritme Haversine.

5.5.1 Perancangan TOPSIS

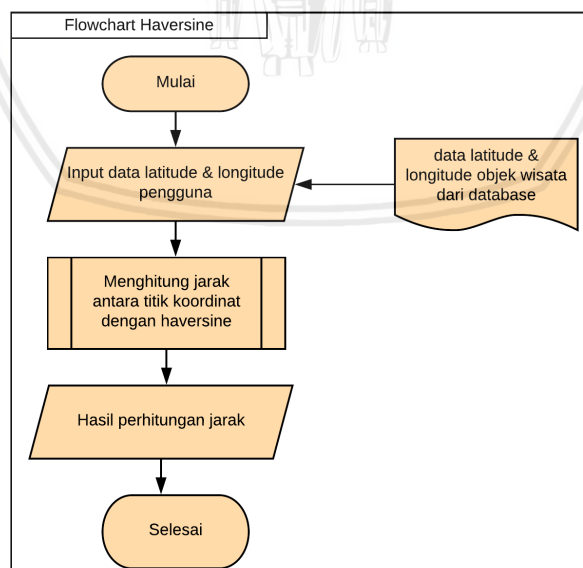
Algoritme TOPSIS dipakai dalam melakukan perhitungan guna mendapatkan rekomendasi objek wisata menurut input kriteria *user*. Gambar 5.6 berikut menjabarkan mengenai tahapan pada proses hitung memakai algoritme TOPSIS. Rincian contoh perhitungan pada setiap langkah langkah akan dijabarkan dalam sub bab 5.5.3.



Gambar 5.6 Flowchart Algoritme TOPSIS

5.5.2 Perancangan Haversine

Haversine ialah algoritme yang digunakan guna mencari Panjang jarak dalam 2 titik koordinat. Tahapan dalam melakukan pencarian jarak dengan metode haversine dijelaskan pada Gambar 5.7. Sedangkan Rincian contoh perhitungan akan dijabarkan dalam sub bab 5.5.3.



Gambar 5.7 Flowchart Algoritme Haversine

5.5.3 Perhitungan Manual

- **Rumus Haversine**

Dengan dasar *flowchart* sebelumnya pada Gambar 5.7, dijabarkan mengenai tahapan perhitungan heversine. Tabel 5.1 menjelaskan *latitude* dan *longitude* pengguna dan tujuannya yang dipakai guna mencari jarak memakai rumus Haversine dalam Persamaan (2.8).

Tabel 5.1 Data Lokasi

Lokasi	Latitude	Longitude
Pengguna	-7,953720	112,614263
Tujuan	-7,957296	112,606141

$$A = \sin^2 \left(\frac{(-7,957296) - (-7,953720)}{2} \right)$$

$$B = \sin^2 \left(\frac{(112,606141) - (112,614263)}{2} \right)$$

$$C = A + \cos(-7,953720) \cos(-7,957296) * B$$

$$d = 2 * 6371 * \arcsin \sqrt{C}$$

$$d = 0,9788$$

Hasil proses hitung yang diperoleh jarak antar 2 titik koordinat sebesar 0, 9788 km.

- **Perhitungan TOPSIS**

a. Memasukkan bobot kriteria dan data alternatif yang dipakai

Tada tabel 5.2 memuat sampel bobot yang dimasukkan, sedangkan data alternatif yang dipakai dalam meranking wisata terdapat di Tabel 5.3.

Tabel 5.2 Tabel bobot kriteria

Kriteria	Bobot
Jarak (J)	9
Harga (H)	9
Jumlah Review (JR)	2
Rating (R)	5

Tabel 5.3 Sampel Data Alternatif

No	Alternatif Wisata	Jarak (J)	Harga (H)	Jumlah Review (JR)	Rating (R)
1	Museum Brawijaya	2,168	3000	2053	43
2	Kampung Jodipan	4,145	3000	9994	43

3	Alun - Alun Malang	3,663	0	18893	44
4	Hawaii Waterpark	5,861	75000	4875	43
5	Malang Night Paradise	5,667	50000	2977	43

b. Menentukan normalisasi matriks alternatif

Normalisasi matriks alternatif diperoleh dengan data alternatif memakai Persamaan (2.1). Mula-mula menghitung jumlah per kriteria dengan cara nilai alternatif pada setiap kriteria dikuadratkan dan dijumlah, setelah itu diakar kuadrat. Kemudian, bagi setiap nilai alternatif dengan jumlah yang telah dihitung. Tabel 5.4 menunjukkan hasil jumlah nilai alternatif setiap kriteria dan Tabel 5.5 menunjukkan perhitungan normalisasi matriks alternatif.

Tabel 5.4 Hasil Jumlah Alternatif Setiap Kriteria

No	Alternatif Wisata	Jarak (J)	Harga (H)	Jumlah Review (JR)	Rating (R)
1	Museum Brawijaya	2,168	3000	2053	43
2	Kampung Jodipan	4,145	3000	9994	43
3	Alun - Alun Malang	3,663	0	18893	44
4	Hawaii Waterpark	5,861	75000	4875	43
5	Malang Night Paradise	5,667	50000	2977	43
Jumlah		10,08796	90238,57	22218,65	96,60228

Berikut merupakan sampel perhitungan di atas.

$$J: \sqrt{2,168^2 + 4,145^2 + 3,663^2 + 5,861^2 + 5,667^2} = 10,08796$$

Tabel 5.5 Hasil Normalisasi Matriks Alternatif

No	Alternatif Wisata	Jarak (J)	Harga (H)	Jumlah Review (JR)	Rating (R)
1	Museum Brawijaya	0,21491	0,033245	0,0924	0,445124
2	Kampung Jodipan	0,410896	0,033245	0,449802	0,445124
3	Alun - Alun Malang	0,363136	0	0,850322	0,455476
4	Hawaii Waterpark	0,580999	0,83113	0,21941	0,445124
5	Malang Night Paradise	0,561739	0,554087	0,133987	0,445124

Berikut merupakan sampel perhitungan di atas..

- Baris 1, Kolom J: $\frac{2,168}{10,087969} = 0,21491$

c. Membuat matriks normalisasi terbobot

Matriks normalisasi terbobot didapatkan menggunakan Persamaan (2.2), dimana hasil normalisasi pada tahap sebelumnya dikalikan bobot dalam Tabel 5.2. Tabel 5.6 berikut merupakan hasil matriks normalisasi terbobot.

Tabel 5.6 Hasil Matriks Normalisasi Terbobot

No	Alternatif Wisata	Jarak (J)	Harga (H)	Jumlah Review (JR)	Rating (R)
1	Museum Brawijaya	1,934187	0,299207	0,1848	2,22562
2	Kampung Jodipan	3,698061	0,299207	0,899605	2,22562
3	Alun - Alun Malang	3,268222	0	1,700643	2,277379
4	Hawaii Waterpark	5,228995	7,480172	0,438821	2,22562
5	Malang Night Paradise	5,05565	4,986781	0,267973	2,22562

Berikut merupakan sampel perhitungan diatas.

- Baris 1, Kolom J: $0,21491 * 9 = 1,934187$

d. Mencari solusi ideal positif dan solusi ideal negatif

Solusi ideal positif juga negatif diperoleh menggunakan Persamaan (2.3) dan (2.4). Apabila kriteria bersifat *benefit* seperti *rating* dan jumlah review, solusi ideal positive adalah nilai paling tinggi kemudian nilai paling rendah merupakan solusi ideal positive. Sebaliknya apabila kriteria bersifat *cost* seperti harga dan jarak nilai solusi ideal positive adalah nilai paling rendah kemudian nilai paling tinggi merupakan solusi ideal negative seperti dijabarkan dalam Tabel 5.7 berikut.

Tabel 5.7 Hasil Solusi Ideal Positive dan Negatif

No	Solusi	J	H	JR	R
1	A+	1,934187	0	1,700643	2,277379
2	A-	5,228995	7,480172	0,1848	2,22562

Sampel pencarian solusi ideal diatas.

- Baris 1, Kolom J:
 $\min\{1,934187; 3,698061; 3,268222; 5,228995; 5,05565\} = 1,934187$
- Baris 2, Kolom J:
 $\max\{1,934187; 3,698061; 3,268222; 5,228995; 5,05565\} = 5,228995$

e. Menghitung Separasi

Untuk menghitung separasi digunakan Persamaan (2.5) dan (2.6). Pertama, jumlahkan kuadrat dari pengurangan nilai normalisasi terbobot per baris dengan solusi ideal positif kemudian akarkan. Selanjutnya jumlahkan kuadrat dari pengurangan nilai normalisasi terbobot setiap baris dengan solusi ideal negatif kemudian akarkan. Hasil proses hitung dijabarkan di Tabel 5.8.

Tabel 5.8 Hasil Perhitungan Separasi

No	Jarak	
	SI+	SI-
1	1,545958	7,90076
2	1,960898	7,377056
3	1,334036	7,880232
4	8,270647	0,254021
5	6,055307	2,500792

Sampel perhitungan ditunjukkan berikut.

- Baris 1, Kolom SI+:

$$\sqrt{(1,934187 - 1,934187)^2 + (0 - 0,299207)^2 + (1,700643 - 0,1848)^2 + (2,277379 - 2,22562)^2} = 1,545958$$

- Baris 1, Kolom SI-:

$$\sqrt{(1,934187 - 5,228995)^2 + (0,299207 - 7,480172)^2 + (0,1848 - 0,1848)^2 + (2,22562 - 2,22562)^2} = 7,90076$$

f. Mengukur kedekatan relatif atas solusi ideal

Untuk mendapatkan kedekatan relatif, digunakan Persamaan (2.7) yaitu nilai solusi ideal negative dibagikan dengan jumlah antara solusi ideal negatif serta positif. Hasil perhitungan dijabarkan di Tabel 5.9.

Tabel 5.9 Hasil Perhitungan Kedekatan Relatif

No	Alternatif Wisata	Preferensi
1	Museum Brawijaya	0,83635
2	Kampung Jodipan	0,790008
3	Alun - Alun Malang	0,855221
4	Hawaii Waterpark	0,029798
5	Malang Night Paradise	0,292282

Sampel perhitungan ditunjukkan berikut.

- Baris 1: $\frac{7,90076}{(1,545958 + 7,90076)} = 0,83635$

g. Merangking Alternatif

Perangkingan preferensi alternative wisata dilakukan melalui pengurutan nilai preferensi alternative mulai yang tinggi hingga yang rendah.

Tabel 5.10 Hasil Perangkingan

No	Alternatif Wisata	Preferensi
1	Alun - Alun Malang	0,855221
2	Museum Brawijaya	0,83635
3	Kampung Jodipan	0,790008
4	Malang Night Paradise	0,292282
5	Hawaii Waterpark	0,029798

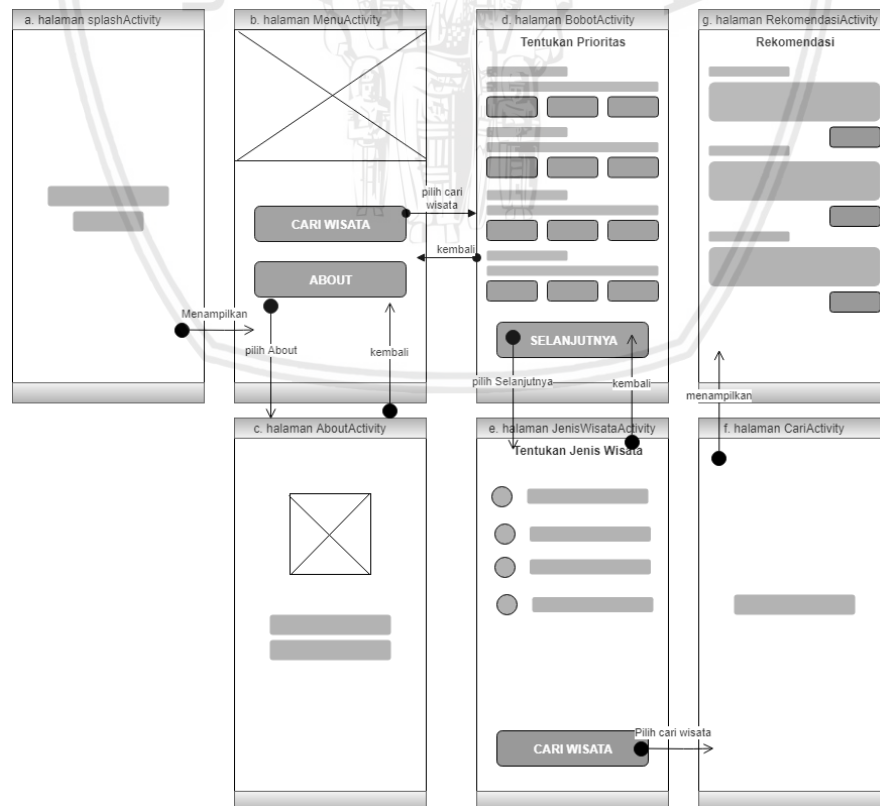
Hasil perangkingan nilai preferensi dijabarkan pada Tabel 5.10. Sehingga dapat diperoleh hasil rekomendasi obyek wisata adalah Alun – Alun Malang dengan nilai 0,855221.

5.6 Perancangan Navigasi dan Antarmuka

Bagian ini akan menjabarkan mengenai perancangan *screenflow* dan tampilan pada aplikasi yang dikembangkan.

5.6.1 Perancangan Screen Flow

Bagian ini merupakan diagram *screenflow* dari halaman aplikasi yang dikembangkan.



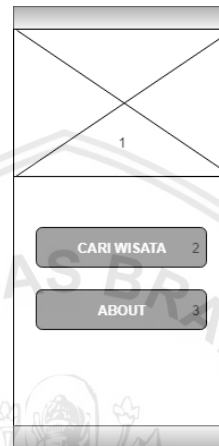
Gambar 5.8 Screenflow Aplikasi

Gambar 5.8 ialah *screenflow* aplikasi yang akan dibangun. Dalam *screen flow* ini memuat total 7 antarmuka mulai dari halaman splashActivity, MenuActivity, AboutActivity, BobotActivity, JenisWisataActivity, CariActivity, hingga Rekomendasi Activity

5.6.2 Perancangan Antarmuka

Bagian ini akan menjabarkan mengenai Rancangan antarmuka aplikasi yang ditampilkan dalam Gambar 5.9 hingga Gambar 5.13.

a. Halaman Menu Utama



Gambar 5.9 Antarmuka Menu Utama

Menu utama yaitu halaman yang terdapat menu guna mengakses fitur cari wisata dan about. Dalam halaman ini terdapat beberapa bagian yaitu:

1. Gambar Cari Wisata Malang
2. Tombol untuk menuju fitur cari wisata
3. Tombol untuk menuju halaman about
- 4.

b. Halaman Bobot Prioritas

Gambar 5.10 Antarmuka Bobot Prioritas

Bobot Prioritas ialah halaman yang dipakai dalam melakukan input data kriteria prioritas. Pada halaman Cari wisata ini terdiri dari 4 kriteria yang harus diisi bobotnya oleh pengguna yaitu jarak, harga, rating, dan jumlah review. Terdapat 3 tingkat bobot kriteria yang disediakan sistem yaitu rendah, sedang, dan tinggi. Bobot rendah bernilai 2, bobot sedang bernilai 5, dan bobot tinggi bernilai 9. Gambar 5.10 merupakan gambar rancangan antarmuka Cari Wisata yang bagiannya yaitu :

1. Menu untuk mengisi bobot Jarak
2. Menu untuk mengisi bobot Harga
3. Menu untuk mengisi bobot Rating
4. Menu untuk mengisi bobot Review
5. Tombol untuk membuka halaman selanjutnya

c. Halaman Jenis Wisata

Jenis Wisata merupakan halaman yang ditampilkan untuk memilih jenis wisata yang diinginkan oleh pengguna seperti pada gambar 5.11 Pada halaman ada beberapa bagian halaman seperti berikut :

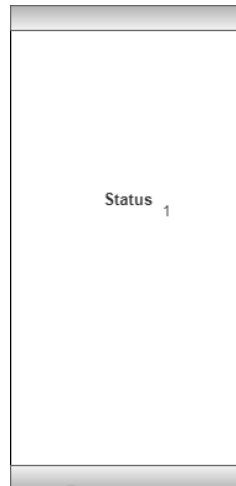
1. *Radio Button* untuk memilih jenis wisata yang diinginkan
2. Tombol untuk melakukan perhitungan

Gambar 5.11 Antarmuka Jenis Wisata

d. Halaman Cari Wisata

Cari Wisata ialah halaman yang muncul saat system sedang melakukan proses perhitungan untuk mendapatkan rekomendasi seperti pada gambar 5.12. yang terdapat bagian yaitu :

1. Text status yang digunakan untuk menampilkan status perhitungan



Gambar 5.12 Antarmuka Cari Wisata

e. Halaman Result / Rekomendasi



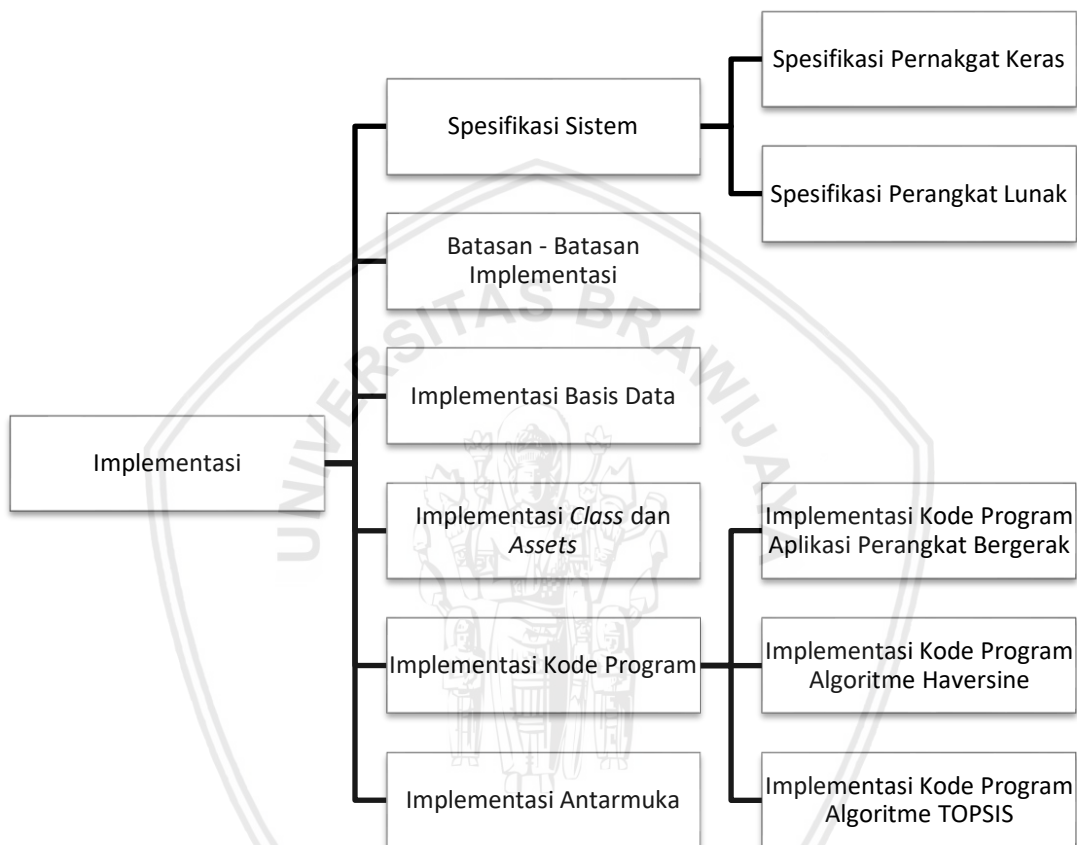
Gambar 5.13 Antarmuka Rekomendasi

Rekomendasi ialah halaman yang dipakai guna menampilkan rekomendasi objek wisata yang telah dihitung oleh sistem. Di dalam halaman rekomendasi terdapat nama tempat wisata beserta alamat wisata dan deskripsi wisata. Selain itu ada juga tombol navigasi yang akan membuka google maps untuk melakukan navigasi. Gambar 5.13 menampilkan rancangan antarmuka yang terdapat bagian yaitu :

1. Nama Tempat Wisata yang direkomendasikan beserta detailnya
2. Tombol navigasi untuk melakukan navigasi menuju tempat wisata

BAB 6 IMPLEMENTASI

Bab ini menjabarkan berkenaan bagaimana penerapan algoritme TOPSIS dalam aplikasi *mobile* rekomendasi wisata dalam kota malang. Tahapan implementasi dijabarkan dalam Gambar 6.1.



Gambar 6.1 Diagram Implementasi

6.1 Spesifikasi Sistem

Pada bab sebelumnya, telah dijelaskan perancangan yang dijadikan dasar pada implementasi. Spesifikasi sistem terbagi 2, yang pertama spesifikasi *hardware*, dan selanjutnya spesifikasi *software*.

6.1.1 Spesifikasi Perangkat Keras

Pada Tabel 6.1 berikut menjabarkan detail *hardware* dipakai pada saat penerapan aplikasi rekomendasi wisata dalam kota Malang.

Tabel 6.1 Spesifikasi *Hardware* Komputer

Komponen	Spesifikasi
Tipe	Komputer Personal
Prosesor	Intel Core i5-3570K 3.40 GHz
Random Access Memory (RAM)	8 GB DDR3
Kartu Grafis	AMD Radeon HD 7770

Sedangkan gawai *mobile* yang digunakan saat pengujian dijelaskan berikut di Tabel 6.2 .

Tabel 6.2 Spesifikasi *Mobile*

Komponen	Spesifikasi
Tipe	Xiaomi Redmi Note 5
Prosesor	Qualcomm Snapdragon 636 @ 1,80 Ghz
Random Access Memory (RAM)	3740 MB
Pengolah Grafis	Adreno 509

6.1.2 Spesifikasi Perangkat Lunak

Tabel 6.3. berikut menjelaskan rincian *software* yang dipakai implementasi aplikasi rekomendasi wisata dalam kota Malang.

Tabel 6.3 Spesifikasi *Software* Komputer

Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Bahasa Pemrograman	Java Android dan PHP
Aplikasi Pengembang	Android Studio, Atom
Aplikasi Server & Database	XAMPP, MySQL
Peramban	Google Chrome

Sedangkan spesifikasi *software* yang dipakai di gawai *mobile* ditampilkan di Tabel 6.4.

Tabel 6.4 Spesifikasi *Software Mobile*

Komponen	Spesifikasi
Sistem Operasi	Android 8.0

6.2 Batasan-Batasan Implementasi

1. Aplikasi diimplementasikan dan hanya kompatibel dengan sistem operasi Android versi 5.0. / lebih baru.
2. Aplikasi ditulis menggunakan JAVA dalam IDE Android Studio.
3. Database yang dipakai sistem ialah MySQL.

6.3 Implementasi Basis Data

Berikut merupakan penerapan *database* memakai phpMyAdmin berupa perintah SQL berikut dalam Kode 6.1.

```
1. CREATE TABLE `wisata` (
2.   `id` int(5) NOT NULL PRIMARY KEY AUTO_INCREMENT,
3.   `nama` varchar(100) NOT NULL
4.   `latitude` decimal(11,8) NOT NULL,
5.   `longitude` decimal(11,8) NOT NULL,
6.   `harga` int(11) NOT NULL,
7.   `rating` int(11) NOT NULL,
8.   `review` int(11) NOT NULL,
9.   `alamat` varchar(255) NOT NULL,
10.  `kategori` varchar(55) NOT NULL,
11. )
```

Kode 6.1 Perintah SQL Membuat Tabel wisata

Kode 6.1 diatas menjabarkan tentang perintah SQL yang digunakan guna mengimplementasi tabel wisata pada sistem. Tabel 6.5 berikut ialah struktur tabel berdasarkan implementasi yang dikerjakan sebelumnya yang dipakai guna menempatkan data objek wisata.

Tabel 6.5 Struktur Tabel Basis Data

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id (PK)	Integer(11)	Guna menempatkan data id	1,2,3, dst
2.	nama	Varchar(100)	Guna menempatkan data nama wisata	Alun Alun Merdeka Malang
3.	latitude	Decimal(11,8)	Guna menempatkan data latitude wisata	-7.98250800
4.	longitude	Decimal(11,8)	Guna menempatkan data longitude wisata	112.63078900
5.	harga	Int(11)	Guna menempatkan data harga wisata	15000
6.	rating	Int(11)	Guna menempatkan data rating wisata	45
7.	review	Int(11)	Guna menempatkan data jumlah review wisata	4196

8.	alamat	Varchar(255)	Guna menempatkan data alamat wisata	Jl. Merdeka Selatan, Kiduldalem, Klojen, Kota Malang
9.	Kategori	Varchar(55)	Guna menempatkan data kategori wisata	Wisata Taman

6.4 Implementasi *Class* dan *Assets Layout*

Berikut merupakan implementasi *Class* dimana berkas Kelas beserta Aset yang diimplementasikan ke dalam program dijabarkan dalam Tabel 6.6. Dimana *file* yang memuat kode pada *activity* memiliki ekstensi .java, sedangkan *layout* dari *activity* memiliki ekstensi .xml.

Tabel 6.6 Berkas Kelas dan *Assets Layout*

No.	Package	Nama Class	Nama Layout
1.	Id.rickyirfandi.cari wisata	SplashActivity.java	activity_splash.xml
2.	Id.rickyirfandi.cari wisata	MenuActivity.java	activity_menu.xml
3.	Id.rickyirfandi.cari wisata	BobotActivity.java	activity_bobot.xml
4.	Id.rickyirfandi.cari wisata	JenisWisataActivity.java	activity_jenis_wisata.xml
5.	Id.rickyirfandi.cari wisata	CariActivity.java	activity_cari.xml
6.	Id.rickyirfandi.cari wisata	RekomendasiActivity.java	activity_rekomendasi.xml
7.	Id.rickyirfandi.cari wisata	Data.java	-
8.	Id.rickyirfandi.cari wisata	TOPSIS.java	-
9.	Id.rickyirfandi.cari wisata	WisataModel.java	-

6.5 Implementasi Kode Program

Bagian ini menjabarkan tentang implementasi pada masing masing kelas di bagian sebelumnya pada Tabel 6.6 menjadi bentuk kode program. Kemudian, terdapat pula implementasi perhitungan jarak dengan menggunakan Algoritme Haversine dan perhitungan rekomendasi wisata dengan menggunakan Algoritme Topsis.

6.5.1 Implementasi Kode Aplikasi Perangkat Bergerak

Berikut ini merupakan kode implementasi kelas *CariActivity* yang memiliki fungsi untuk mendapatkan hasil rekomendasi wisata. Implementasi berikut dibangun berlandaskan *class diagram* yang telah dirancang sebelumnya.

```

1  public class CariActivity extends AppCompatActivity {
2      TextView status;
3      List<WisataModel> listWisata;
4      private LocationRequest mLocationRequest;
5
6      private long UPDATE_INTERVAL = 10 * 10000;
7      private long FASTEST_INTERVAL = 10 * 10000;
8      private static final int REQUEST_LOCATION_PERMISSION = 1;
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_cari);
14
15         status = findViewById(R.id.textView4);
16
17         startLocationUpdates();
18     }
19
20     private void BuatListWisata() {
21         String JSON = Data.JSON_Data;
22         listWisata = new ArrayList<>();
23
24         try {
25             JSONArray wisata = new JSONArray(JSON);
26             for (int i = 0; i < wisata.length(); i++) {
27                 JSONObject temp = wisata.getJSONObject(i);
28                 String id = temp.getString("id");
29                 String nama = temp.getString("nama");
30                 String alamat = temp.getString("alamat");
31                 String kategori = temp.getString("kategori");
32                 String foto = temp.getString("foto");
33                 double latitude = temp.getDouble("latitude");
34                 double longitude = temp.getDouble("longitude");
35                 double jarak = Jarak(Data.latitude,
36                 Data.longitude, latitude, longitude);
37
38                 int harga = temp.getInt("harga");
39                 int rating = temp.getInt("rating");
40                 int review = temp.getInt("review");
41
42                 if (kategori.equalsIgnoreCase("Wisata Taman")) {
43                     if (Data.WISATA_TAMAN) {
44                         WisataModel tmp = new
45                         WisataModel(id, nama, alamat, kategori, foto, latitude, longitude, jarak,
46                         harga, rating, review);
47                         listWisata.add(tmp);
48                     }
49                 } else if (kategori.equalsIgnoreCase("Wisata
50 Museum")) {
51                     if (Data.WISATA_MUSEUM) {
52                         WisataModel tmp = new
53                         WisataModel(id, nama, alamat, kategori, foto, latitude, longitude, jarak,
54                         harga, rating, review);
55                         listWisata.add(tmp);
56                     }
57                 } else if (kategori.equalsIgnoreCase("Wisata
58

```

```

59     Kolan")) {
60         if(Data.WISATA_KOLAM) {
61             WisataModel tmp = new
62 WisataModel(id,nama,alamat,kategori,foto,latitude,longitude,jarak,
63 harga,rating,review);
64             listWisata.add(tmp);
65         }
66     } else if (kategori.equalsIgnoreCase("Wisata
67 Foto")) {
68         if(Data.WISATA_FOTO) {
69             WisataModel tmp = new
70 WisataModel(id,nama,alamat,kategori,foto,latitude,longitude,jarak,
71 harga,rating,review);
72             listWisata.add(tmp);
73         }
74     } else {
75     }
76     }
77 }
78
79 for(int i = 0; i < listWisata.size(); i++){
80     String nama = listWisata.get(i).nama;
81 }
82
83 status.setText("Mencari Rekomendasi");
84
85 } catch (Exception e) {
86     e.printStackTrace();
87 }
88
89 }
90
91
92
93 protected void startLocationUpdates() {
94     mLocationRequest = new LocationRequest();
95
96 mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURAC
97 Y);
98     mLocationRequest.setInterval(UPDATE_INTERVAL);
99     mLocationRequest.setFastestInterval(FATEST_INTERVAL);
100
101     LocationSettingsRequest.Builder builder = new
102 LocationSettingsRequest.Builder();
103     builder.addLocationRequest(mLocationRequest);
104     LocationSettingsRequest locationSettingsRequest =
105 builder.build();
106
107     SettingsClient settingsClient =
108 LocationServices.getSettingsClient(this);
109
110 settingsClient.checkLocationSettings(locationSettingsRequest);
111
112
113     if (ActivityCompat.checkSelfPermission(this,
114 Manifest.permission.ACCESS_FINE_LOCATION) !=
115 PackageManager.PERMISSION_GRANTED &&
116 ActivityCompat.checkSelfPermission(this,
117 Manifest.permission.ACCESS_COARSE_LOCATION) !=
118 PackageManager.PERMISSION_GRANTED) {
119         ActivityCompat.requestPermissions(this, new String[]
120 {Manifest.permission.ACCESS_FINE_LOCATION,
121     REQUEST_LOCATION_PERMISSION});
122     } else {

```

```

122 getFusedLocationProviderClient(this).requestLocationUpdates(mLocat
123 ionRequest, new LocationCallback() {
124     @Override
125     public void
126 onLocationResult(LocationResult locationResult) {
127
128     Location location = locationResult.getLastLocation();
129     Data.latitude = location.getLatitude();
130     Data.longitude = location.getLongitude();
131
132     BuatListWisata();
133
134
135
136     TOPSIS topsis = new TOPSIS(listWisata);
137     WisataModel[] hasil;
138     hasil = topsis.HitungTOPSIS();
139     Intent intent = new Intent(CariActivity.this,
140 RekomendasiActivity.class);
141     Data.HasilRekomendasi = hasil;
142     startActivity(intent);
143     finish();
144     }
145 },
146 Looper.myLooper());
147 }
148 }
149 }
150 }
151 }
152 }
153 }

```

Kode 6.2 Kode Class CariActivity

Kode 6.2 diatas ialah penerapan kode program kelas guna memperoleh rekomendasi wisata. Berikut penjabaran kode kelas *CariActivity*, yaitu:

1. Baris 11-19, merupakan perintah *onCreate* merupakan perintah yang pertama kali dijalankan saat activity dijalankan. Di dalam perintah ini, memanggil fungsi.
2. Baris 21-92, merupakan fungsi *BuatListWisata*. Di dalam fungsi ini akan membuka seluruh objek wisata yang ada dalam JSON Array untuk dicek apakah objek wisata tersebut termasuk dengan jenis wisata yang diinginkan user. Apabila termasuk, objek tersebut akan masuk ke dalam *ListWisata* dalam bentuk *WisataModel*. Apabila tidak, maka aplikasi akan mengecek object wisata selanjutnya.
3. Baris 93-123, merupakan fungsi *startLocationUpdate*. Di dalam fungsi ini akan dilakukan pengambilan koordinat pengguna untuk digunakan dalam perhitungan.
4. Baris 125-135, merupakan fungsi *onLocationResult* yang akan dipanggil saat koordinat pengguna telah didapatkan. Dimana data latitude beserta longitude disimpan ke dalam variabel.
5. Baris 138 – 155, merupakan kode bagian dari *onLocationResult* yang digunakan untuk melakukan perhitungan TOPSIS dengan memakai kelas TOPSIS. Setelah perhitungan selesai dilakukan, selanjutnya memanggil activity *HasilRekomendasi*.

6.5.2 Implementasi Kode Algoritme Haversine

Algoritme Haversine merupakan sebuah algoritme yang berfungsi untuk mendapatkan nilai jarak pada antar dua titik koordinat. Implementasi kode algoritme Haversine dijabarkan seperti berikut.

```

1  public double Jarak(double lat1, double lon1, double lat2, double
2  lon2){
3      int RADIUS BUMI = 6371;
4      double dLat = Math.toRadians(lat2 - lat1);
5      double dLon = Math.toRadians(lon2 - lon1);
6      double initialLat = Math.toRadians(lat1);
7      double finalLat = Math.toRadians(lat2);
8
9      double a = Math.sin(dLat/2) * Math.sin(dLat/2);
10     double b = Math.sin(dLon/2) * Math.sin(dLon/2);
11     double c = a + Math.cos(initialLat) * Math.cos(finalLat) * b;
12     double d = 2 * RADIUS BUMI * Math.asin(Math.sqrt(c)) ;
13     return d;
14 }

```

Kode 6.4 Kode Algoritme Haversine

Kode 6.4 merupakan penerapan kode mencari jarak dengan algoritme Haversine. Penjabarannya, yaitu:

1. Baris 1-2, ialah deklarasi fungsi guna menghitung jarak dengan menggunakan algoritme haversine.
2. Baris 3-7, ialah deklarasi variabel yang dipakai untuk perhitungan algoritme haversine.
3. Baris 9 -13, merupakan perhitungan algoritme haversine.

6.5.3 Implementasi Kode Algoritme TOPSIS

Algoritme TOPSIS merupakan algoritme yang dipakai dalam penghitungan rekomendasi wisata Malang. Dalam implementasi dijabarkan kode dari algoritme TOPSIS yang dibangun seperti yang telah dirancang dalam bagian sebelumnya.

```

1  public class TOPSIS {
2      private List<WisataModel> listWisata;
3      private WisataModel[] listHasil;
4      private double PEMBAGI_JARAK, PEMBAGI_HARGA, PEMBAGI_REVIEW,
5      PEMBAGI_RATING;
6      private double[][] MATRIX_NORMALISASI;
7      private double[][] MATRIX_TERBOBOT;
8      private double[] Aplus = new double[4];
9      private double[] Amin = new double[4];
10     private double[] SIplus;
11     private double[] SImin;
12     private double[] JumlahSI;
13     private double[] CIPlus;
14
15     public TOPSIS(List<WisataModel> listWisata) {
16         this.listWisata = listWisata;
17     }
18
19     public WisataModel[] HitungTOPSIS() {
20         HITUNG_PEMBAGI();
21         HITUNG_NORMALISASI();
22         HITUNG_TERBOBOT();

```

```

23      HitungAPlusAmin();
24      HitungSIPlusSIMin();
25      HitungCIPlus();
26      BuatRekomendasi();
27      return listHasil;
28  }
29
30  //STEP 1 : mendapatkan pembagi dari masing masing kriteria
31  private void HITUNG_PEMBAGI () {
32      double jarak = 0;
33      int harga = 0, review = 0, rating = 0;
34      for(int i = 0; i < listWisata.size(); i++){
35          jarak += (listWisata.get(i).getJarak() *
36 listWisata.get(i).getJarak());
37          harga += (listWisata.get(i).getHarga() *
38 listWisata.get(i).getHarga());
39          review += (listWisata.get(i).getReview() *
40 listWisata.get(i).getReview());
41          rating += (listWisata.get(i).getRating() *
42 listWisata.get(i).getRating());
43      }
44      PEMBAGI_JARAK = Math.sqrt(jarak);
45      PEMBAGI_HARGA = Math.sqrt(harga);
46      PEMBAGI_REVIEW = Math.sqrt(review);
47      PEMBAGI_RATING = Math.sqrt(rating);
48  }
49
50
51  //STEP 2 : Menghitung tabel normalisasi
52  private void HITUNG_NORMALISASI () {
53
54      MATRIX_NORMALISASI = new double [listWisata.size()][4];
55      for(int i = 0; i < listWisata.size(); i++){
56          for (int j = 0; j < 4; j++) {
57              switch(j) {
58                  case 0:
59                      double jarak =
60 listWisata.get(i).getJarak();
61                      jarak = jarak/PEMBAGI_JARAK;
62                      MATRIX_NORMALISASI[i][j] = jarak;
63                      break;
64                  case 1:
65                      double harga =
66 listWisata.get(i).getHarga();
67                      harga = harga/PEMBAGI_HARGA;
68                      MATRIX_NORMALISASI[i][j] = harga;
69                      break;
70                  case 2:
71                      double review =
72 listWisata.get(i).getReview();
73                      review = review/PEMBAGI_REVIEW;
74                      MATRIX_NORMALISASI[i][j] = review;
75                      break;
76                  case 3:
77                      double rating =
78 listWisata.get(i).getRating();
79                      rating = rating/PEMBAGI_RATING;
80                      MATRIX_NORMALISASI[i][j] = rating;
81                      break;
82                  default:
83                      break;
84              }
85          }
86      }
87  }

```

```

89
90 //STEP 3 : Menghitung tabel terbobot
91 private void HITUNG_TERBOBOT() {
92     MATRIX_TERBOBOT = new double [listWisata.size()][4];
93     int BOBOT_JARAK = Data.BOBOT_JARAK,
94         BOBOT_HARGA = Data.BOBOT_HARGA,
95         BOBOT_RATING = Data.BOBOT_RATING,
96         BOBOT_REVIEW = Data.BOBOT_REVIEW;
97
98     for(int i = 0; i < listWisata.size(); i++){
99         for (int j = 0; j < 4; j++ ){
100             switch(j) {
101                 case 0:
102                     double jarak = MATRIX_NORMALISASI[i][j];
103                     jarak = jarak * BOBOT_JARAK;
104                     MATRIX_TERBOBOT[i][j] = jarak;
105                     break;
106                 case 1:
107                     double harga = MATRIX_NORMALISASI[i][j];
108                     harga = harga * BOBOT_HARGA;
109                     MATRIX_TERBOBOT[i][j] = harga;
110                     break;
111                 case 2:
112                     double review = MATRIX_NORMALISASI[i][j];
113                     review = review * BOBOT_REVIEW;
114                     MATRIX_TERBOBOT[i][j] = review;
115                     break;
116                 case 3:
117                     double rating = MATRIX_NORMALISASI[i][j];
118                     rating = rating * BOBOT_RATING;
119                     MATRIX_TERBOBOT[i][j] = rating;
120                     break;
121                 default:
122                     }
123             }
124         }
125
126 //STEP 4 : Menghitung nilai A+ dan A- dari setiap kriteria
127 // A+ Jarak : MIN, A+ Harga : MIN, A+ Review : MAX, A+ Review
128 : MAX
129 // A- Jarak : MAX, A+ Harga : MAX, A+ Review : MIN, A+ Review
130 : MIN
131 public void HitungAplusAmin(){
132     double Aplus_Jarak = 9999, Aplus_Harga = 9999,
133     Aplus_Review = 0, Aplus_Rating = 0;
134     double Amin_Jarak = 0, Amin_Harga = 0, Amin_Review = 9999,
135     Amin_Rating = 9999;
136     for(int i = 0; i < listWisata.size(); i++){
137         for (int j = 0; j < 4; j++ ){
138             switch(j) {
139                 case 0:
140                     double jarak = MATRIX_TERBOBOT[i][j];
141                     if(jarak < Aplus_Jarak){
142                         Aplus_Jarak = jarak;
143                     }
144                     if(jarak > Amin_Jarak){
145                         Amin_Jarak = jarak;
146                     }
147                     break;
148                 case 1:
149                     double harga = MATRIX_TERBOBOT[i][j];
150                     if(harga < Aplus_Harga){
151                         Aplus_Harga = harga;
152                     }
153                     if(harga > Amin_Harga){
154                         Amin_Harga = harga;
155                     }
156                 }
157             }
158         }
159     }

```

```

156         if(harga>Amin_Harga){
157             Amin_Harga = harga;
158         }
159         break;
160     case 2:
161         double review = MATRIX_TERBOBOT[i][j];
162         if(review>Aplus_Review){
163             Aplus_Review = review;
164         }
165         if(review<Amin_Review){
166             Amin_Review = review;
167         }
168         break;
169     case 3:
170         double rating = MATRIX_TERBOBOT[i][j];
171         if(rating>Aplus_Rating){
172             Aplus_Rating = rating;
173         }
174         if(rating<Amin_Rating){
175             Amin_Rating = rating;
176         }
177         break;
178     default:
179     }
180 }
181 }
182 Aplus[0]=Aplus_Jarak;
183 Aplus[1]=Aplus_Harga;
184 Aplus[2]=Aplus_Review;
185 Aplus[3]=Aplus_Rating;
186
187 Amin[0]=Amin_Jarak;
188 Amin[1]=Amin_Harga;
189 Amin[2]=Amin_Review;
190 Amin[3]=Amin_Rating;
191 }
192
193
194 //STEP 5 : Menghitung nilai SI+ dan SI- dari setiap kriteria
195 Juga Jumlah SI
196 public void HitungSIPlusSIMin(){
197     SIplus = new double[listWisata.size()];
198     SImin = new double[listWisata.size()];
199     JumlahSI = new double[listWisata.size()];
200     CIPlus = new double[listWisata.size()];
201
202     //SI+
203     for(int i = 0; i < listWisata.size(); i++){
204         double SIPlus_ = ((Aplus[0] -
205 MATRIX_TERBOBOT[i][0])*(Aplus[0] - MATRIX_TERBOBOT[i][0]));
206         SIPlus_ += ((Aplus[1] -
207 MATRIX_TERBOBOT[i][1])*(Aplus[1] - MATRIX_TERBOBOT[i][1]));
208         SIPlus_ += ((Aplus[2] -
209 MATRIX_TERBOBOT[i][2])*(Aplus[2] - MATRIX_TERBOBOT[i][2]));
210         SIPlus_ += ((Aplus[3] -
211 MATRIX_TERBOBOT[i][3])*(Aplus[3] - MATRIX_TERBOBOT[i][3]));
212         SIPlus_ = Math.sqrt(SIPlus_);
213         SIplus[i] = SIPlus_;
214     }
215
216     //SI-
217     for(int i = 0; i < listWisata.size(); i++){
218         double SImin_ = ((MATRIX_TERBOBOT[i][0] -
219 Amin[0])*(MATRIX_TERBOBOT[i][0] - Amin[0]));
220         SImin_ += ((MATRIX_TERBOBOT[i][1] -
221 Amin[1])*(MATRIX_TERBOBOT[i][1] - Amin[1]));

```

```

222         SMin_ += ((MATRIX_TERBOBOT[i][2] -
223 Amin[2])*(MATRIX_TERBOBOT[i][2] - Amin[2]));
224         SMin_ += ((MATRIX_TERBOBOT[i][3] -
225 Amin[3])*(MATRIX_TERBOBOT[i][3] - Amin[3]));
226         SMin_ = Math.sqrt(SMin_);
227         SMin[i] = SMin_;
228     }
229
230     //Jumlah SI
231     for(int i = 0; i < listWisata.size(); i++){
232         double jumlah = (SIplus[i]+SMin[i]);
233         JumlahSI [i] = jumlah;
234     }
235 }
236
237 //Step 6 Menghitung nilai CI+
238 private void HitungCIPlus(){
239     for(int i = 0; i < listWisata.size(); i++){
240         double CIplus_ = (SMin[i]/JumlahSI[i]);
241         CIPlus[i] = CIplus_;
242     }
243 }
244
245 //Step 7 : Mangambil top 3 Rekomendasi
246 private void BuatRekomendasi(){
247     double[][] TOP3 = new double[3][2]; //menyimpan int urutan
248     alternatif pada list
249     listHasil = new WisataModel[3];
250
251     for(int i = 0; i < listWisata.size(); i++){
252         if(CIPlus[i] > TOP3[0][0]){
253             TOP3[2][0] = TOP3[1][0];
254             TOP3[2][1] = TOP3[1][1];
255             TOP3[1][0] = TOP3[0][0];
256             TOP3[1][1] = TOP3[0][1];
257             TOP3[0][0] = CIPlus[i];
258             TOP3[0][1] = i;
259         } else if (CIPlus[i] > TOP3[1][0]){
260             TOP3[2][0] = TOP3[1][0];
261             TOP3[2][1] = TOP3[1][1];
262             TOP3[1][0] = CIPlus[i];
263             TOP3[1][1] = i;
264         } else if (CIPlus[i] > TOP3[2][0]){
265             TOP3[2][0] = CIPlus[i];
266             TOP3[2][1] = i;
267         }
268     }
269     WisataModel temp = listWisata.get((int)TOP3[0][1]);
270     WisataModel temp2 = listWisata.get((int)TOP3[1][1]);
271     WisataModel temp3 = listWisata.get((int)TOP3[2][1]);
272     listHasil[0] = temp;
273     listHasil[1] = temp2;
274     listHasil[2] = temp3;
275 }

```

Kode 6.5 Kode Kelas *CariActivity*

Kode 6.5 diatas ialah penerapan kode kelas TOPSIS dalam menghitung rekomendasi wisata. Penjelasan potongan kode diatas, yaitu:

1. Baris 1-13, ialah deklarasi variabel variabel yang akan dipakai dalam perhitungan TOPSIS
2. Baris 15-17, adalah konstruktor dari kelas TOPSIS

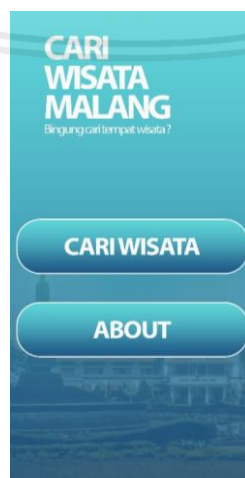
3. Baris 19-28, merupakan fungsi HitungTOPSIS yang berisi urutan pemanggilan fungsi menghitung TOPSIS dan mengembalikan array WisataModel yang berisi hasil rekomendasi wisata.
4. Baris 31-49, merupakan fungsi untuk menghitung pembagi dari setiap kriteria dengan menjumlahkan dan mengakarkannya.
5. Baris 52-87, merupakan fungsi menghitung tabel normalisasi dimana setiap nilai dibagi dengan pembaginya di setiap kriteria.
6. Baris 91-122, merupakan fungsi untuk menghitung table terbobot. Pada fungsi ini dilakukan pengalihan setiap nilai dengan bobot setiap kriteria yang telah diinputkan oleh pengguna sebelumnya
7. Baris 129-192, merupakan fungsi mendapatkan nilai A+ dan A- dari setiap kriteria. Dimana pada cari wisata ini A+ jarak dan harga diambil nilai yang terkecil sedangkan A+ review dan rating diambil nilai yang terbesar. Sebaliknya, A- Jarak dan Harga diambil nilai terbesar, dan A- review dan rating diambil nilai terkecil
8. Baris 196-235, merupakan fungsi untuk mendapatkan SI+ dan SI-. Setelah keduanya telah didapatkan, juga dilakukan perhitungan untuk mendapatkan jumlah SI
9. Baris 238-242, merupakan fungsi untuk menghitung nilai CI+ yang didapatkan dengan membagi nilai SI- dengan jumlah SI
10. Baris 245-274, merupakan fungsi untuk mendapatkan nilai CI+ 3 teratas yang nantinya akan keluar sebagai hasil rekomendasi

6.6 Implementasi Antarmuka

Bagian ini menunjukkan antarmuka yang diterapkan dengan dasar rancangan Gambar 6.11 hingga 6.14 dan hasilnya akan dijabarkan dalam Gambar 6.2 hingga Gambar 6.6

a. Menu Utama

Antarmuka Menu Utama diimplementasikan di Gambar 6.2 berikut dengan dasar perancangan sebelumnya pada Gambar 5.9.



Gambar 6.2 Hasil Implementasi Menu Utama

b. Bobot Prioritas

Antarmuka Bobot Prioritas diimplementasikan di Gambar 6.3 berikut dengan dasar perancangan sebelumnya pada Gambar 5.10.



Langkah 1 : Tentukan Prioritas

Jarak : **TINGGI**
Semakin tinggi prioritas, maka sistem akan mencari wisata yang lebih dekat

Harga : **TINGGI**
Semakin tinggi prioritas, maka sistem akan mencari wisata yang lebih murah

Rating : **SEDANG**
Semakin tinggi prioritas, maka sistem akan mencari wisata yang memiliki rating tinggi

Review : **RENDAH**
Semakin tinggi prioritas, maka sistem akan mencari wisata yang memiliki review banyak

SELANJUTNYA

Gambar 6.3 Hasil Implementasi Bobot Prioritas

c. Jenis Wisata

Antarmuka Jenis Wisata diimplementasikan di Gambar 6.4 berikut dengan dasar perancangan sebelumnya pada Gambar 5.11.



Langkah 2 : Pilih Jenis Wisata

☒ Wisata Taman

☐ Wisata Museum

☐ Wisata Kolam Renang

☐ Wisata Foto

☐ Semua Wisata

CARI WISATA

Gambar 6.4 Hasil Implementasi Jenis Wisata

d. Cari Wisata

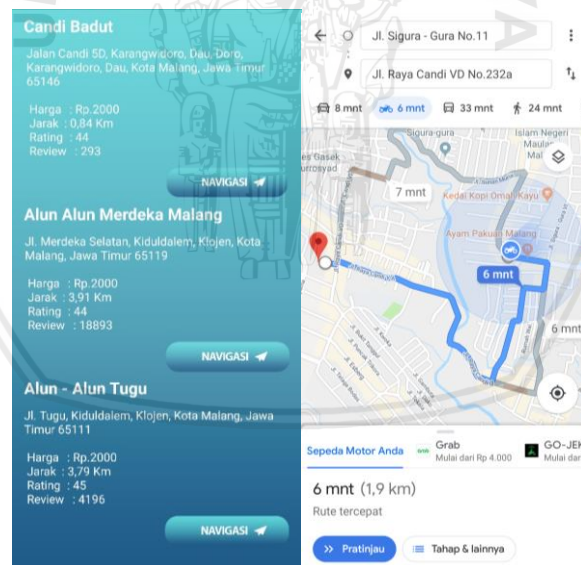
Antarmuka Cari Wisata diimplementasikan di Gambar 6.5 berikut dengan dasar perancangan sebelumnya pada Gambar 5.12



Gambar 6.5 Hasil Implementasi Cari Wisata

e. Rekomendasi

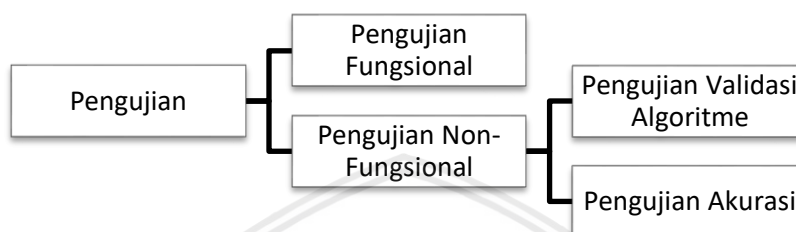
Antarmuka Rekomendasi diimplementasikan di Gambar 6.6 berikut dengan dasar perancangan sebelumnya pada Gambar 5.13.



Gambar 6.6 Hasil Implementasi Rekomendasi

BAB 7 PENGUJIAN

Bagian ini akan menjelaskan mengenai pengujian yang dikerjakan terhadap aplikasi yang telah diterapkan sebelumnya. Pengujian yang dipakai ialah pengujian fungsional juga non-fungsional seperti dijelaskan Gambar 7.1 berikut.



Gambar 7.1 Diagram Pengujian

7.1 Pengujian Fungsional

Pengujian fungsional merupakan pengujian dengan tujuan dalam memeriksa aplikasi yang sedia dibangun memenuhi keperluan yang diinginkan. Pengujian fungsional ini, digunakan pengujian *black box testing* berdasarkan daftar kebutuhan sebelumnya.

Pada pengujian fungsional, hanya akan menguji aliran utama saja pada skenario *usecase* dimana keadaan koneksi internet tersambung dengan baik dan GPS hidup. Aliran alternatif dimana koneksi internet terputus dan GPS tidak dalam keadaan aktif tidak dilakukan pengujian disebabkan terdapat dalam batasan masalah di sub bab 1.5 yang disebutkan status koneksi internet juga sensor GPS pada perangkat bergerak diasumsikan dalam kondisi yang baik, karena itu pengujian aliran alternatif tidak perlu dilakukan. Selain itu, dalam pengujian akurasi dibutuhkan koneksi internet yang tersambung juga GPS yang menyala, sehingga pengujian aliran alternatif dapat diasumsikan tidak signifikan.

Pada pengujian fungsional yang akan diuji, dibutuhkan sebuah kasus uji. Kasus uji fungsional yang akan dilakukan dapat dilihat pada Tabel 7.1

Tabel 7.1 Kasus Uji Fungsional Mendapatkan Rekomendasi Wisata

Nomor Kasus Uji	KUF-001
Nama Kasus Uji	Mendapatkan Rekomendasi Wisata
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Guna membuktikan aplikasi yang dikembangkan telah berhasil melakukan perhitungan rekomendasi wisata dan menampilkan kepada pengguna
Prosedur Pengujian	1. Jalankan aplikasi.

	2. Memilih Cari Wisata. 3. Menginputkan data bobot prioritas pada form <i>input</i> yang telah disediakan. 4. Memilih Selanjutnya. 5. Memasukkan data jenis Wisata yang ingin dicari 6. Memilih Cari Wisata.
Hasil yang Diharapkan	<i>User</i> mendapatkan hasil rekomendasi wisata berdasarkan data yang telah diinputkan.

Dari kasus uji diatas, dilakukan percobaan sesuai prosedur dan diperoleh hasil pengujian fungsional seperti dalam tabel 7.2 berikut dengan hasil bernilai valid 100%.

Tabel 7.2 Hasil Pengujian Fungsional

Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
KUF-001	<i>User</i> mendapatkan hasil rekomendasi wisata dengan dasar data yang dimasukkan sebelumnya.	Aplikasi berhasil memunculkan hasil rekomendasi wisata berdasarkan data yang telah diinputkan.	Valid

7.2 Pengujian Non-Fungsional

Pengujian non-fungsional dikerjakan dengan melakukan uji validasi algoritme dan pengujian akurasi seperti tertera dalam Tabel 4.5.

7.2.1 Pengujian Validasi Algoritme

Validasi algoritme dikerjakan melalui perbandingan antara urutan rekomendasi wisata yang dikeluarkan oleh aplikasi dan manual sama. Pengujian Validasi Algoritme pada metode TOPSIS dianggap valid apabila rekomendasi wisata dari aplikasi dan manual urutannya identik (Nurrachman, 2016) . Pada Tabel 7.3 berikut menjabarkan tentang data alternatif yang dipakai.

Tabel 7.3 Data Alternatif

No	Alternatif Wisata	Jarak (J)	Harga (H)	Jumlah Review (JR)	Rating (R)
1	Museum Brawijaya	2,168	3000	2053	43
2	Kampung Jodipan	4,145	3000	9994	43
3	Alun - Alun Malang	3,663	0	18893	44
4	Hawaii Waterpark	5,861	75000	4875	43
5	Malang Night Paradise	5,667	50000	2977	43

Tabel 7.4 Perpadanan Rekomendasi Aplikasi dan Manual

No	Hasil Aplikasi		Hasil Manual	
	Urutan	Nilai	Urutan	Nilai
1.	Alun - Alun Malang	0,8553	Alun - Alun Malang	0,8552
2.	Museum Brawijaya	0,8363	Museum Brawijaya	0,8363
3.	Kampung Jodipan	0,7900	Kampung Jodipan	0,7900
4.	Malang Night Paradise	0,2922	Malang Night Paradise	0,2922
5.	Hawaii Waterpark	0,0297	Hawaii Waterpark	0,0297

Tabel 7.4 menjabarkan perpadanan hasil rekomendasi aplikasi dan manual yang didapat dari sub bab 5.1.5.3. Dari tabel tersebut, diperoleh sebuah urutan rekomendasi dari perhitungan sistem dan manual. Tabel 7.5 berikut menjabarkan tentang kecocokan urutan rekomendasi.

Tabel 7.5 Hasil Kecocokan Validasi Algoritme

No	Hasil Rekomendasi		Kecocokan
	Aplikasi	Manual	
1.	Alun - Alun Malang	Alun - Alun Malang	Valid
2.	Museum Brawijaya	Museum Brawijaya	Valid
3.	Kampung Jodipan	Kampung Jodipan	Valid
4.	Malang Night Paradise	Malang Night Paradise	Valid
5.	Hawaii Waterpark	Hawaii Waterpark	Valid

Berdasarkan pengujian tersebut dapat diperoleh hasil bernilai valid dikarenakan baik hasil perhitungan sistem juga manual menunjukkan hasil rekomendasi yang identik. Sehingga metode TOPSIS sukses diterapkan dengan baik.

7.2.2 Pengujian Akurasi

Bagian ini dikerjakan dengan melakukan uji coba terhadap 30 pakar berbeda dengan menggunakan kriteria dan alternatif yang berbeda-beda sesuai dengan keinginan. Pakar yang diwawancarai dalam penelitian ini merupakan orang yang rata rata dalam satu bulan melakukan kunjungan wisata di Malang minimal 2 kali atau lebih. Orang yang memenuhi kriteria tersebut dianggap telah memiliki pengetahuan tempat wisata di Malang yang cukup sehingga dapat dianggap pakar.

Tabel 7.6 berikut menjabarkan tentang hasil keluaran rekomendasi aplikasi yang dikembangkan di penelitian ini dan kesesuaiannya dengan keinginan pakar.

Tabel 7.6 Hasil Pengujian Akurasi

No	ID User	ID Alternatif Wisata	Hasil Rekomendasi	
			Sistem	Pakar
1.	User 1	1, 8, 13, 17, 18, 19, 26, 27	1, 8, 19	1, 8, 19
2.	User 2	1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12, 3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 4, 23, 24, 25	3, 16, 6	3, 16, 6
3.	User 3	4, 23, 24, 25	23, 25, 24	23, 25, 24
4.	User 4	4, 23, 24, 25	23, 24, 25	23, 24, 25
5.	User 5	1, 8, 13, 17, 18, 19, 26, 27, 4, 23, 24, 25	24, 4, 1	Kolam Renang Gajayana
6.	User 6	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 1, 8, 13, 17, 18, 19, 26, 27	3, 16, 8	3, 16, 8
7.	User 7	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 2, 5, 7, 12	3, 2, 6	3, 2, 6
8.	User 8	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 2, 5, 7, 12	3, 2, 6	3, 2, 6
9.	User 9	1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12	2, 1, 8	2, 1, 8
10.	User 10	4, 23, 24, 25	24, 23, 25	24, 23, 25
11.	User 11	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 2, 5, 7, 12	3, 2, 6	3, 2, 6
12.	User 12	1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12, 3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 4, 23, 24, 25	3, 2, 6	4
13.	User 13	1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12	2, 1, 8	2, 1, 8
14.	User 14	2, 5, 7, 12	2, 7, 12	2, 7, 12
15.	User 15	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 2, 5, 7, 12	3, 16, 2	3, 16, 2
16.	User 16	1, 8, 13, 17, 18, 19, 26, 27, 4, 23, 24, 25	24, 4, 1	24, 4, 1

17.	User 17	4, 23, 24, 25	4, 24, 23	4, 24, 23
18.	User 18	1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12	2, 1, 17	2, 1, 17
19.	User 19	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30	3, 16, 6	3, 16, 6
20.	User 20	1, 8, 13, 17, 18, 19, 26, 27	1, 17, 8	1, 17, 8
21.	User 21	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12	16, 17, 3	16, 17, 3
22.	User 22	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30	16, 21, 30	16, 21, 30
23.	User 23	2, 5, 7, 12, 4, 23, 24, 25	2, 7, 24	2, 7, 24
24.	User 24	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30	3, 16, 6	14
25.	User 25	1, 8, 13, 17, 18, 19, 26, 27, 4, 23, 24, 25	4, 24, 1	4, 24, 1
26.	User 26	1, 8, 13, 17, 18, 19, 26, 27, 4, 23, 24, 25	4, 1, 24	23
27.	User 27	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12	3, 2, 6	1
28.	User 28	3, 6, 9, 10, 11, 14, 15, 16, 20, 21, 22, 28, 29, 30, 2, 5, 7, 12	3, 2, 16	3, 2, 16
29.	User 29	2, 5, 7, 12	2, 5, 7	2, 5, 7
30.	User 30	1, 8, 13, 17, 18, 19, 26, 27, 2, 5, 7, 12	2, 1, 17	2, 1, 17

Dalam uji akurasi yang sudah dikerjakan, maka bisa didapatkan nilai akurasi dengan menghitung jumlah rekomendasi yang sesuai dengan keinginan pakar.

Tabel 7.7 Hasil Kecocokan Pengujian Akurasi

No	Hasil Rekomendasi		Kecocokan
	Sistem	Pakar	
1	1, 8, 19	1, 8, 19	v
2	3, 16, 6	3, 16, 6	v
3	23, 25, 24	23, 25, 24	v
4	23, 24, 25	23, 24, 25	v

5	24, 4, 1	Kolam Renang Gajayana	x
6	3, 16, 8	3, 16, 8	v
7	3, 2, 6	3, 2, 6	v
8	3, 2, 6	3, 2, 6	v
9	2, 1, 8	2, 1, 8	v
10	24, 23, 25	24, 23, 25	v
11	3, 2, 6	3, 2, 6	v
12	3, 2, 6	4	x
13	2, 1, 8	2, 1, 8	v
14	2, 7, 12	2, 7, 12	v
15	3, 16, 2	3, 16, 2	v
16	24, 4, 1	24, 4, 1	v
17	4, 24, 23	4, 24, 23	v
18	2, 1, 17	2, 1, 17	v
19	3, 16, 6	3, 16, 6	v
20	1, 17, 8	1, 17, 8	v
21	16, 17, 3	16, 17, 3	v
22	16, 21, 30	16, 21, 30	v
23	2, 7, 24	2, 7, 24	v
24	3, 16, 6	14	x
25	4, 24, 1	4, 24, 1	v
26	4, 1, 24	23	x
27	3, 2, 6	1	x
28	3, 2, 16	3, 2, 16	v
29	2, 5, 7	2, 5, 7	v
30	2, 1, 17	2, 1, 17	v

Tabel 7.7 merupakan tabel yang menampilkan hasil pengujian antara hasil rekomendasi wisata oleh sistem dan rekomendasi oleh pakar. Ketika hasil rekomendasi yang dikeluarkan sistem sesuai dengan rekomendasi pakar, maka pada kolom kecocokan bernilai valid (v). Sebaliknya apabila tidak sama dengan rekomendasi pakar, maka akan bernilai tidak valid (x).

Berdasarkan pengujian dalam Tabel 7.7, diperoleh data uji bernilai valid sesuai dengan rekomendasi pakar wisata sejumlah 25 data, kemudian yang bernilai tidak

valid sebanyak 5 data. Maka dapat dihitung akurasi metode TOPSIS dengan melakukan pembagian jumlah percobaan yang sesuai keinginan pakar dengan seluruh percobaan seperti berikut:

$$\text{Nilai Akurasi} = \frac{25}{30} \times 100\% = \frac{2500\%}{30} = 83,33\%$$

Berdasarkan hasil perhitungan akurasi di atas, maka didapatkan nilai akurasi dari rekomendasi wisata dalam kota Malang dengan metode TOPSIS adalah sebesar 83,33%.



BAB 8 KESIMPULAN DAN SARAN

8.1 Kesimpulan

Berdasarkan hasil dari penelitian yang dikerjakan sebelumnya, diperoleh kesimpulan sebagai berikut:

1. Perancangan dikerjakan menjadi beberapa tahap. Yang pertama ialah melakukan identifikasi aktor dan analisis untuk mendapatkan kebutuhan fungsional dan non fungsional. Setelah itu, dilakukan perancangan perangkat lunak dengan merancang activity diagram, sequence diagram, class diagram, basis data. Setelah itu tahap selanjutnya adalah perancangan algoritme. Perancangan algoritme terdiri dari perancangan algoritme TOPSIS, algoritme dan Haversine. Tahap terakhir adalah Melakukan perancangan Navigasi dan Antarmuka. Pada tahap ini dimulai dengan melakukan perancangan Screen Flow dan terakhir dilakukan perancangan antarmuka. Setelah semua selesai, rancangan tersebut siap untuk diimplementasikan.
2. Sistem rekomendasi wisata dalam kota Malang berhasil diterapkan memakai Android Studio Java. Sistem juga diimplementasikan dengan metode TOPSIS dan data alternatif berupa data objek wisata, dan kriteria yang digunakan merupakan jarak wisata, harga, *rating*, dan jumlah review. Bobot prioritas untuk setiap kriteria diinput oleh *user* dengan 3 tingkatan, yaitu: rendah, sedang dan tinggi. Dimana tingkatan rendah bernilai 2, sedang bernilai 5, dan tinggi bernilai 9. Pencarian rekomendasi alternatif dihitung memakai algoritme TOPSIS. Data alternatif yang terpilih adalah alternatif dengan nilai preferensi tertinggi dan akan ditampilkan dalam bentuk list yang berisi data wisata yang terdiri dari nama wisata, alamat, jarak, harga, *rating*, hingga jumlah review. Terdapat pula tombol navigasi pada setiap wisata rekomendasi untuk melakukan navigasi dari koordinat user saat ini menuju wisata yang diinginkan dengan memakai Google Maps.
3. Pengujian dikerjakan dengan Pengujian Fungsional beserta Non-Fungsional. Pengujian fungsional dilakukan guna mengecek fungsi sistem berjalan dengan baik dan dalam penelitian ini menghasilkan nilai valid. Pengujian Non-Fungsional dibagi menjadi dua yaitu uji Validasi Algoritme dan Pengujian Akurasi. Pengujian Validasi Algoritme dikerjakan untuk mencocokkan hasil perhitungan manual dengan keluaran system yang berupa urutan hasil rekomendasi wisata. Pada pengujian Validasi Algoritme digunakan 5 data alternatif yang hasilnya menunjukkan hasil urutan rekomendasi wisata yang sama antara perhitungan manual dengan keluaran system. Hasil pengujian akurasi antara hasil metode TOPSIS, dan keputusan pakar dilakukan sebanyak 30 kali dengan pakar yang berbeda. Hasil metode TOPSIS diperoleh diperoleh 25 data percobaan sesuai dengan keinginan pakar dengan tingkat akurasi sebesar 83,33%.

8.2 Saran

Saran yang dapat dibagi guna pengembangan lebih lanjut dari penelitian ini, antara lain:

1. Perlu tambahan metode lain dalam menetapkan bobot sebuah kriteria yang dipakai. Pada penelitian ini bobot masih dimasukkan secara manual oleh pengguna.
2. Perlu ditambahkan kriteria alternatif lain agar system dapat melakukan pencarian rekomendasi wisata yang lebih akurat. Pada penelitian ini hanya memakai 4 kriteria yaitu jarak, harga, rating, dan jumlah review.



DAFTAR REFERENSI

- Indarwasti, A. 2017. EATGASM: Sistem Informasi Berbasis Website Sebagai Sistem Pendukung Keputusan Pemilihan Tempat Kuliner Di Depok Menggunakan Metode Topsis. *MULTINETICS*, [e-journal] 3(1) Tersedia di: <<http://jurnal.pnj.ac.id/index.php/multinetics/article/view/1114>> [Diakses 30 Agustus 2018]
- Kozina, Y., Volkova, N., Horpenko, D. 2018. Mobile Application for Decision Support in Multi-Criteria Problems. *IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, [e-journal] pp. 56-59 Tersedia di: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8478499&isnumber=8478415>> [Diakses 30 Agustus 2018]
- Kurniasih, D. L. 2017. Sistem pendukung keputusan pemilihan laptop dengan metode TOPSIS. *Pelita Informatika: Informasi dan Informatika*, [e-journal] 3(2) Tersedia di: <<http://ejurnal.stmik-budidarma.ac.id/index.php/pelita/article/view/142>> [Diakses 30 Agustus 2018]
- Nurrachman, R. 2016, Implementasi Metode AHP-TOPSIS Pada Sistem Rekomendasi Kuliner. Malang: Universitas Brawijaya.
- Octavia, D. 2017. Aplikasi Pencarian Tempat Futsal dengan Menggunakan Metode Topsis. [online] Tersedia di: <http://simki.unpkediri.ac.id/mahasiswa/file_artikel/2017/12.1.03.02.0319.pdf> [Diakses 30 Agustus 2018]
- Purnama, B. E. 2011. Pemanfaatan Global Positioning System Untuk Pelacakan Objek Bergerak. *Speed-Sentra Penelitian Engineering dan Edukasi*, [e-journal] 1(4) Tersedia di: <<http://ijns.org/journal/index.php/speed/article/view/848>> [Diakses 30 Agustus 2018]
- Purnomo, E. N. S., Sihwi, S. W., & Anggrainingsih, R. 2016. Analisis Perbandingan Menggunakan Metode AHP, TOPSIS, dan AHP-TOPSIS dalam Studi Kasus Sistem Pendukung Keputusan Penerimaan Siswa Program Akselerasi. *ITSMART: Jurnal Teknologi dan Informasi*, [e-journal] 2(1) Tersedia di: <<https://jurnal.uns.ac.id/itsmart/article/view/612>> [Diakses 30 Agustus 2018]
- Sasongko, D. 2018. Wisata Kota Malang siap sumbang peningkatan kunjungan wisatawan, [onlinel] Tersedia di: <<https://malang.merdeka.com/kabar-malang/wisata-kota-malang-siap-sumbang-peningkatan-kunjungan-wisatawan-170807r.html>> [Diakses 30 Agustus 2018]
- Satyaputra, A., Aritonang, E., M., 2016, Let`s Build Your Android Apps with Android Studio. Jakarta: Elex Media Komputindo

- Setiawan, H., 2016. Implementasi Haversine Formula Pada Lokasi Pariwisata Berbasis *Smartphone*. *Jurnal Fahma*, [e-journal] 14(2) Tersedia di: <<http://portal.ejurnal.net/index.php/fahma/article/view/1334>> [Diakses 30 Agustus 2018].
- Setiawan, I., Andjarwirawan, J., & Handojo, A. 2013. Aplikasi makassar tourism pada kota makassar berbasis android. *Jurnal Infra*, [e-journal] pp. 156 Tersedia di: <<http://publication.petra.ac.id/index.php/teknikinformatika/article/view/797>> [Diakses 30 Agustus 2018].
- Sutanta, E., Mustofa, H. 2012. Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov Di Pemkab Bantul Yogyakarta. *JURNAL TEKNOLOGI INFORMASI & KOMUNIKASI (JURTIK)* 1.1, [e-journal] Tersedia di: <https://repository.ugm.ac.id/33043/1/2012_MAKALAH_08_JURNAL_JURTIK_Edhy_Sutanta_Khabib_Mustofa_Kebutuhan_Web_Service_Untuk_Sinkronisasi_Data_Antar_SISFO_dlm_e-Gov_di_Pemkab_Bantul_Yogyakarta_0.pdf> [Diakses 30 Agustus 2018]



LAMPIRAN A

A.1 Data Yang Digunakan

ID Alternatif	Nama Tempat	Harga	Jumlah Review	Rating	Latitude	Longitude	Alamat
1	Museum Brawijaya	3000	2053	43	-7.972.086	112.621.224	Jl. Ijen No.25 A, Gading Kasri, Klojen, Kota Malang, Jawa Timur 65115
2	Kampung Jodipan	3000	9994	43	-7.983.225	112.637.631	Gang 1 Jodipan, Kesatrian, Blimbing, Kota Malang, Jawa Timur 65126
3	Alun - Alun Malang	0	17839	44	-7.982.500	112.630.807	Jl. Merdeka Selatan, Kiduldalem, Klojen, Kota Malang, Jawa Timur 65119
4	Hawaii Waterpark	75000	4875	43	-7.923.448	112.658.239	Perumahan Graha Kencana Raya, Jl. Raya Karanglo, Karanglo, Malang, Jawa Timur 65126
5	Malang Night Paradise	50000	2977	43	-7.924.098	112.656.542	Jl. Graha Kencana Raya No.66, Karanglo, Balarjosari, Blimbing, Kota Malang, Jawa Timur 65126
6	Alun - Alun Tugu	0	4196	45	-7.977.136	112.634.060	Jl. Tugu, Kiduldalem, Klojen, Kota Malang, Jawa Timur 65111
7	Kampung Tridi	3000	497	43	-7.981.829	112.638.111	Jl. Temenggungan Ledok, Kesatrian, Blimbing, Kota Malang, Jawa Timur 65121
8	Candi Badut	0	293	44	-7.957.724	112.598.605	Jalan Candi 5D, Karangwidoro, Dau, Doro, Karangwidoro, Dau, Kota Malang, Jawa Timur 65146
9	Taman Wisata Wendit	15000	2812	39	-7.952.761	112.673.989	Jl. Raya Wendit Tim., Lowoksoro, Mangliawan, Pakis, Malang, Jawa Timur 65154

10	Brawijaya Edupark	15000	952	40	-7.975.521	112.631.287	Jl. Kahuripan No.1, Klojen, Kota Malang, Jawa Timur 65119
11	Taman Rekreasi Kota Malang	500	486	39	-7.978.838	112.633.319	Jalan Simpang, Jl. Mojopahit No.1, Kiduldalem, Klojen, Kota Malang, Jawa Timur 65111
12	Wisma Tumapel	5000	83	39	-7.978.221	112.632.849	Jalan Tumapel No.1, Kiduldalem, Klojen, Kauman, Klojen, Kota Malang, Jawa Timur 65119
13	Museum Malang Tempoe Doeloe	25000	137	39	-7.978.909	112.634.931	Jalan Gajahmada, Kiduldalem, Klojen, Kiduldalem, Klojen, Kota Malang, Jawa Timur 65119
14	Slamet City Park	0	1992	43	-7.973.689	112.622.454	Jl. Taman Slamet No.8, Gading Kasri, Klojen, Kota Malang, Jawa Timur 65115
15	Malabar Park	0	675	42	-7.968.482	112.626.518	Jl. Malabar, Oro-oro Dowo, Klojen, Kota Malang, Jawa Timur 65119
16	Taman Singha Merjosari	0	3077	43	-7.945.143	112.603.229	Jl. Mertojoyo Selatan Blk. B No.20, Merjosari, Kec. Lowokwaru, Kota Malang, Jawa Timur 65144
17	Museum Mpu Purwa	0	151	43	-7.940.328	112.620.848	Jl. Soekarno Hatta Perumahan Griya Samta Blk. B No.210, Mojolangu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141
18	Museum Musik Indonesia	0	198	43	-7.988.308	112.626.647	Jalan Nusakambangan No.19, Kasin, Klojen, Kasin, Klojen, Kota Malang, Jawa Timur 65117
19	Museum Bendoel Prima	0	224	43	-7.986.765	112.632.059	Jl. Wiromargo, No. 32, 65137, Sukoharjo, Klojen, Kota Malang, Jawa Timur 65118
20	Merbabu Family Park	0	133	43	-7.967.954	112.625.828	Jl. Merbabu, Oro-oro Dowo, Klojen, Kota Malang, Jawa Timur 65119
21	Taman Kunang Kunang	0	846	42	-7.963.151	112.623.650	Jalan Jakarta, Oro-oro Dowo, Klojen, Penanggungan, Klojen, Kota Malang, Jawa Timur 65113

22	Taman Trunojoyo	0	873	43	-7.976.401	112.636.514	Jl. Trunojoyo, Klojen, Kota Malang, Jawa Timur 65111
23	Kolam Renang Lembah Dieng	15000	1365	43	-7.965.187	112.598.619	Jalan Wisata No.99, Sumberjo, Kalisongo, Dau, Malang, Jawa Timur 65151
24	Taman Rekreasi Sengkaling	25000	3071	42	-7.915.228	112.587.185	Jl. Raya Mulyoagung No.188, Mulyoagung Dau, Sengkaling, Jetis, Mulyoagung, Dau, Malang, Jawa Timur 65153
25	Taman Rekreasi Tlogomas	30000	728	40	-7.930.664	112.604.523	Jalan Baiduri Pandan No.17, Tlogomas, Lowokwaru, Tlogomas, Kec. Lowokwaru, Kota Malang, Jawa Timur 65144
26	Museum Zoologi Frater Vianney	20000	60	43	-7.958.583	112.596.202	Jalan Karang Widoro No.7, Doro, Karangwidoro, Dau, Malang, Jawa Timur 65146
27	Indonesian Old Cinema Museum	0	19	46	-7.937.316	112.627.711	Jl. Soekarno Hatta No.45, Mojolangu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65142
28	UNMER Malang Park	0	32	38	-7.973.238	112.612.850	Jl. Terusan Dieng, Pisang Candi, Sukun, Kota Malang, Jawa Timur 65146
29	Taman Nivea Family Care	0	7	37	-7.967.502	112.625.889	Jl. Merbabu No.18, Oro-oro Dowo, Klojen, Kota Malang, Jawa Timur 65119
30	Taman Idjen Boulevard	0	62	46	-7.968.507	112.623.664	Jl. Besar Ijen, Oro-oro Dowo, Klojen, Kota Malang, Jawa Timur 65115

A.2 Data User

No	ID User	Nama User	Email
1.	User 1	Vevina Dyahsasi Nugraha	vevinadyah@gmail.com
2.	User 2	Wahyu Alif Kurniawan	wahyualif88@gmail.com
3.	User 3	Dahru Wildan Pradanaputra	wildandahru@gmail.com
4.	User 4	Andre Diofanu	andrediofanu@gmail.com
5.	User 5	Tantri Prabandari	tprabandari@gmail.com
6.	User 6	Wahyu Eko	wahyu38eko@gmail.com
7.	User 7	Dodi Indrawicaksono	dodiindra72@gmail.com
8.	User 8	Mashudi	Mashudi1223@gmail.com
9.	User 9	Arda Satata	sandal502@gmail.com
10.	User 10	Joang Pratama Achmad	joangpratama27@gmail.com
11.	User 11	Hafshah	fhah03@gmail.com
12.	User 12	Agmarzah Kurnia	hawkermech99@gmail.com
13.	User 13	Enggar pratama kurniawan	itsenggar@gmail.com
14.	User 14	Amilia	amiliarosa97@gmail.com
15.	User 15	Zanuramdan Dwi Saputra	justzanu24@gmail.com
16.	User 16	Yosua Silalahi	yosilahi@yahoo.com
17.	User 17	Nurina ardy garini	nurinaardyagarini@gmail.com
18.	User 18	Heru Apriadi	heruapr@gmail.com
19.	User 19	Abdurrahman Prawira purmiaji	abdurrahmanprawira01@gmail.com
20.	User 20	Ade Darmawan	Darmawan.Ade98@yahoo.com
21.	User 21	Windi Rahmawati	windirahmawati19@gmail.com
22.	User 22	Muhammad Hidayatullah	mhidayatullah50@gmail.com
23.	User 23	Ary Muhammad Prayoga	arymp111@gmail.com
24.	User 24	Manggala Rezka Perdana	manggala.rezka@gmail.com
25.	User 25	Andy	andyindra27@gmail.com
26.	User 26	David Adi Prasetya	david.a.prasetya@gmail.com

27.	User 27	Lintang A	lintangam12@gmail.com
28.	User 28	Mukhammad Saiful Anwar	mukhammadsaifulanwar@gmail.com
29.	User 29	Sarah Meirina	sarah.meirinasari2@yahoo.com
30.	User 30	Mayovio	mayovio.vio@gmail.com

