IMPLEMENTASI TOPSIS PADA SISTEM REKOMENDASI TEMPAT WISATA PANTAI DI SEKITAR MALANG BERBASIS LOKASI

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhamad Hilmi Hibatullah
NIM: 155150200111098

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

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

IMPLEMENTASI TOPSIS PADA SISTEM REKOMENDASI TEMPAT WISATA PANTAI DI SEKITAR MALANG BERBASIS LOKASI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhamad Hilmi Hibatullah
NIM: 155150200111098

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing 2
Ratih Kartika Dewi, S.T., M.Kom	Drs. Marji, M.T
NIK: 201503 890520 2 001	NIP: 19670801 199203 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

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

repository.ub.ac.id

PERYATAAN 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, 3 Januari 2019

Muhamad Hilmi Hibatullah

NIM: 155150200111098

UNIVERSITAS BRAWIJAYA

PRAKATA

Puji syukur kami panjatkan kehadirat Allah Yang Maha Esa, karena hanya atas limpahan dan rahmat Karunia-Nya lah, penulis dapat menyelesaikan skripsi dengan judul "Implementasi TOPSIS Pada Sistem Rekomendasi Tempat Wisata Pantai Di Sekitar Malang Berbasis Lokasi " ini dengan lancar.

Penyusunan skripsi tak lepas dari bantuan secara moril yang berupa bimbingan, kritik, saran, dukungan, motivasi maupun doa banyak pihak. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada :

1. Ibu Ratih Kartika Dewi, S.T.,M.Kom selaku dosen pembimbing I dan Bapak Drs.Marji, M.T selaku dosen pembimbing II, atas segala bimbingan dan waktu yang telah diluangkan serta kritik dan saran yang telah diberikan kepada penulis.
2. Bapak Agus Wahyu Widodo S.T, M.Sc selaku Ketua Program Studi Teknik Informatika dan dosen pembimbing akademik yang telah memberikan pengarahan selama penulis menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua Jurusan Teknik Infromatika.
4. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini
5. Seluruh civitas academica Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian skripsi ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat secara langsung atau tidak langsung agar terselesaikannya skripsi ini.

Penulis mengucapkan terima kasih atas segala bantuan yang telah diberikan. Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca dan untuk pengembangan selanjutnya. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan oleh keterbatasan kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, 3 Januari 2019

Penulis

Email: hilmihibatullah@gmail.com

repository.ub.ac.id

ABSTRAK

Muhamad Hilmi Hibatullah. 2018. : Implementasi TOPSIS Pada Sistem Rekomendasi Tempat Wisata Pantai Di Sekitar Malang Berbasis Lokasi. Skripsi Program Studi Informatika/Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Ratih Kartika Dewi, S.T., M.Kom. Dan Drs.Marji, M.T.

Sektor pariwisata di Indonesia saat ini dinilai efektif peranannya dalam menambah devisa negara. Pada tahun 2016 saja, sektor pariwisata merupakan penyumbang devisa negara terbesar kedua dengan jumlah USD12,5 miliar. Malang sendiri merupakan salah satu daerah yang memiliki potensi objek wisata yang banyak, dimana salah satunya adalah objek wisata pantai yang cukup banyak. Wisata pantai yang terdapat di daerah Malang cukup banyak dimana tidak jarang menyebabkan pengunjung bingung untuk memilih pantai yang akan dikunjungi. Penelitian ini mengusulkan dibuatnya suatu sistem rekomendasi pantai di sekitar Malang dengan menerapkan metode *Technique For Order Preference by Similiarity to Ideal Solution* (TOPSIS). Metode TOPSIS dipilih karena TOPSIS mampu melakukan perangkingan alternatif terpilih, dimana alternatif terpilih mempunyai jarak terdekat dari solusi ideal positif dan jarak terjauh dari jarak solusi ideal negatif. Kriteria yang digunakan pada sistem ini adalah jarak antara pengguna dan lokasi pantai, biaya masuk pantai, rating, fasilitas yang ditawarkan dan transportasi yangbisa digunakan untuk menuju lokasi pantai. Dalam pengujian fungsionalitas, peneliti menggunakan metode blackbox, dimana didapatkan hasil bahwa 100% dari fungsionalitas sudah valid. Selanjutnya pengujian validasi algoritma, dimana pengujian dilakukan dengan membandingkan hasil keluaran sistem dengan perhitungan manual dan didapatkan hasil 100% sama. Terakhir adalah pengujian rank consistency untuk mengetahui tingkat konsistensi dari hasil rekomendasi dengan menambahkan dan mengurangi kriteria. Hasil yang didapatkan pada pengujian rank consistency menunjukan bahwa metode TOPSIS memiliki tingkat konsistensi yang baik pada saat di implementasikan pada sistem rekomendasi pantai

Kata Kunci: sistem pendukung keputusan, pantai, topsis, rank consistency

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Muhamad Hilmi Hibatullah. 2018. : Implementasi TOPSIS Pada Sistem Rekomendasi Tempat Wisata Pantai Di Sekitar Malang Berbasis Lokasi. Skripsi Program Studi Informatika/Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Ratih Kartika Dewi, S.T., M.Kom. Dan Drs.Marji, M.T.

The tourism sector in Indonesia currently considered to be an effective role in increasing the country's foreign exchange. In 2016, the tourism sector was the second largest constributor to foreign exchange in the amount of USD12.5 billion. Malang is one of the areas that have a lot of potential tourism object. Beach tourism in Malang is a lot where it is cause visitors confused to choose the beach to be visited. This research propose to make a beach recommendation system around Malang by applying the Technique For Order Preference by Similiarity to Ideal Solution (TOPSIS) method. The TOPSIS method was chosen because TOPSIS was able to do the selected alternative ranking, where the selected alternative had the closest distance from the positive ideal solution and the farthest distance from the negative ideal solution. The criteria used in this system are distance, cost, rating, facilities and transportation. System functional testing shows that 100% of the functionality is valid. Furthermore, testing the validation of the algorithm, obtained the results 100% of similarities between the system output and manual calculation. While rank consistency testing obtained results that show that the TOPSIS method has a good level of consistency when implemented on the beach recommendation system.

Keywords: *decision support system, beach, topsis, rank consistency.*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

IMPLEMENTASI TOPSIS PADA SISTEM REKOMENDASI TEMPAT WISATA PANTAI DI SEKITAR MALANG BERBASIS LOKASI i
PENGESAHAN ii
PERYATAAN ORISINALITAS iii
PRAKATA iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL x
DAFTAR GAMBAR xii
DAFTAR KODE xiv
DAFTAR PERSAMAAN xv
DAFTAR LAMPIRAN xvi
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 3
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan masalah 3
1.6 Sistematika pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN 6
2.1 Penelitian Terdahulu 6
2.2 Sistem Pendukung Keputusan 7
2.2.1 Faktor Utama Sistem Pendukung Keputusan 7
2.3 Metode *Technique For Order Preference by Similarity to Ideal Solution* (TOPSIS) 8
2.3.1 Langkah-langkah TOPSIS 8
2.4 Android 10
2.4.1 Arsitekrut Android 11
2.4.2 *Android Software Development Kit* (SDK) 14

2.5 Pengujian Perangkat Lunak 14
2.5.1 ISO 9126 14
2.5.2 Pengujian Black Box 15
2.5.3 Pengujian *Rank Consistency* 16
BAB 3 METODOLOGI 17
3.1 Studi Literatur 17
3.2 Pengumpulan Data 17
3.3 Analisis Kebutuhan 18
3.4 Perancangan Sistem 18
3.5 Implementasi Sistem 18
3.6 Pengujian Sistem 18
3.6.1 Pengujian Black Box 18
3.6.2 Pengujian Validasi Algoritma 18
3.6.3 Pengujian *Rank Consistency* 19
3.7 Pengambilan Kesimpulan dan Saran 19
BAB 4 ANALISIS DAN PERANCANGAN 20
4.1 Analisis Kebutuhan 21
4.1.1 Gambaran Aplikasi 21
4.1.2 Identifikasi Aktor 22
4.1.3 Analisis Kebutuhan Fungsional 22
4.1.4 Analisis Kebutuhan Non Fungsional 27
4.2 Perancangan Perangkat Lunak 27
4.2.1 Perancangan *Activity Diagram* 28
4.2.2 Perancangan Arsitektur Sistem 30
4.2.3 Perancangan *Sequence Diagram* 32
4.2.4 Perancangan *Class Diagram* 35
4.2.5 Perancangan Basis Data 37
4.2.6 Perancangan Algoritma 37
4.2.7 Perancangan Navigasi dan Antarmuka 43
BAB 5 IMPLEMENTASI 49
5.1 Spesifikasi Sistem 49
5.1.1 Spesifikasi Perangkat Keras 49

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.1.2 Spesifikasi Perangkat Lunak 50
5.2 Batasan-Batasan Implementasi 50
5.3 Implementasi Basis Data 51
5.4 Implementasi *Class* dan *Assets* pada File Program 55
5.5 Implementasi Kode Program 56
5.5.1 Implementasi Kode Program Aplikasi Perangkat Bergerak 56
5.5.2 Implementasi Kode Program *Web Service* 61
5.5.3 Implementasi Kode Program Algoritma TOPSIS 65
5.6 Implementasi Antarmuka 66
BAB 6 PENGUJIAN DAN ANALISIS 69
6.1 Pengujian Fungsional 69
6.2 Pengujian Non-Fungsional 72
6.2.1 Pengujian Validasi Algoritma 72
6.2.2 Pengujian *Rank Consistency* 73
6.3 Analisis 76
6.3.1 Analisis Pengujian Fungsional 76
6.3.2 Analisis Pengujian Non-Fungsional 77
BAB 7 KESIMPULAN DAN SARAN 79
7.1 Kesimpulan 79
7.2 Saran 79
DAFTAR PUSTAKA 80
LAMPIRAN 82

DAFTAR TABEL

Tabel 2.1 Penelitian Rujukan 6
Tabel 4.1 Identifikasi Aktor 22
Tabel 4.2 Spesifikasi Kebutuhan Fungsional 22
Tabel 4.3 Spesifikasi *Use Case* Menampilkan Daftar Pantai 23
Tabel 4.4 Spesifikasi *Use Case* Mendapatkan Rekomendasi 24
Tabel 4.5 Spesifikasi *Use Case* Menampilkan Informasi Pantai 24
Tabel 4.6 Spesifikasi *Use Case* Menampilkan Peta 24
Tabel 4.7 Skenario *Use Case* Menampilkan Daftar Pantai 24
Tabel 4.8 Skenario *Use Case* Memasukkan Bobot Kriteria 25
Tabel 4.9 Skenario *Use Case* Mendapatkan Rekomendasi 25
Tabel 4.10 Skenario *Use Case* Menampilkan Informasi Pantai 26
Tabel 4.11 Skenario *Use Case* Menampilkan Peta 27
Tabel 4.12 Spesifikasi Kebutuhan Non Fungsional 27
Tabel 4.13 Sampel Bobot Kriteria 39
Tabel 4.14 Sampel Data Alternatif 39
Tabel 4.15 Hasil Penjumlahan Alternatif Tiap Kriteria 39
Tabel 4.16 Hasil Normalisasi Matriks Alternatif 40
Tabel 4.17 Hasil Normaliasi Matriks Terbobot 40
Tabel 4.18 Hasil Solusi Ideal Positif dan Solusi Ideal Negatif 41
Tabel 4.19 Hasil Perhitungan Jarak Terhadap Solusi Ideal Positif dan Negatif 41
Tabel 4.20 Hasil Perhitungan Nilai Prefrensi 42
Tabel 4.21 Hasil Perangkingan 42
Tabel 5.1 Spesifikasi Perangkat Keras Komputer 50
Tabel 5.2 Spesifikasi Perangkat Bergerak 50
Tabel 5.3 Spesifikasi Perangkat Lunak pada Komputer 50
Tabel 5.4 Spesifikasi Perangkat Lunak pada Perangkat Bergerak 50
Tabel 5.5 Tabel Basis Data *table*_fasilitas 52
Tabel 5.6 Tabel Basis Data *table*_kriteria 53
Tabel 5.7 Tabel Basis Data *table_location* 53
Tabel 5.8 Tabel Basis Data *table*_pantai 53

Tabel 5.9 Tabel Basis Data *table*_relasi_fasilitas 54
Tabel 5.10 Tabel Basis Data *table*_relasi_transportasi 54
Tabel 5.11 Tabel Basis Data *table*_transportasi 54
Tabel 5.12 Berkas Kelas dan *Assets Layout* pada Aplikasi Perangkat Bergerak 55
Tabel 6.1 Kasus Uji Fungsional Menampilkan Daftar Pantai 69
Tabel 6.2 Kasus Uji Fungsional Memasukkan Bobot Kriteria 70
Tabel 6.3 Kasus Uji Melihat Hasil Rekomendasi 70
Tabel 6.4 Kasus Uji Fungsional Menampilkan Informasi Pantai 71
Tabel 6.5 Kasus Uji Fungsional Menampilkan Peta 71
Tabel 6.6 Bobot Kriteria 72
Tabel 6.7 Data Alternatif 72
Tabel 6.8 Perbandingan Keluaran Sistem dan Perhitungan Manual 73
Tabel 6.9 Data Alternatif 74
Tabel 6.10 Hasil Pengujian Fungsional 76
Tabel 6.11 Hasil Pengujian Validasi Algoritma 77
Tabel 6.12 Hasil Pengujian *Rank Consistency* (Rekomendasi) 77

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Android 12
Gambar 2.2 Struktur hierarki ISO 9126 15
Gambar 3.1 AlurcMetodologi 17
Gambar 4.1 Diagram Pohon Perancangan 20
Gambar 4.2 Gambaran Umum 21
Gambar 4.3 *Use Case Diagram* Kebutuhan Fungsional 23
Gambar 4.4 *Activity Diagram* Menampilkan Daftar Pantai 28
Gambar 4.5 *Activity Diagram* Memasukkan Bobot Kriteria 29
Gambar 4.6 *Activity Diagram* Melihat Hasil Rekomendasi 29
Gambar 4.7 *Activity Diagram* Menampilkan Informasi Pantai 30
Gambar 4.8 *Activity Diagram* Menampilkan Peta 30
Gambar 4.9 Arsitektur Perangkat Bergerak 31
Gambar 4.10 Akses Data Eksternal JSON Daftar Pantai 31
Gambar 4.11 Akses Data Eksternal JSON Hasil Rekomendasi Pantai 32
Gambar 4.12 *Sequence Diagram* Menampilkan Daftar Pantai 33
Gambar 4.13 *Sequence Diagram* Memasukan Bobot Kriteria 33
Gambar 4.14 *Sequence Diagram* Melihat Hasil Rekomendasi 34
Gambar 4.15 *Sequence Diagram* Menampilkan Informasi Pantai 34
Gambar 4.16 *Sequence Diagram* Menampilkan Peta 35
Gambar 4.17 *Class Diagram* 36
Gambar 4.18 Perancangan Basis Data 37
Gambar 4.19 *Flowchart* TOPSIS 38
Gambar 4.20 *Screen Flow* Aplikasi Perangkat Bergerak 43
Gambar 4.21 Antarmuka Halaman *Home* 44
Gambar 4.22 Antarmuka Halaman *Input* Bobot Kriteria 45
Gambar 4.23 Antarmuka Halaman Hasil Rekomendasi 46
Gambar 4.24 Antarmuka Halaman Detail Pantai 47
Gambar 4.25 Antarmuka Halaman Map dan Rute 48
Gambar 5.1 Diagram Pohon Implementasi Sistem 49
Gambar 5.2 Hasil Implementasi Basis Data 55

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 5.3 Hasil Implementasi Antarmuka *Home* 67
Gambar 5.4 Hasil Implementasi Antarmuka *Input* Bobot Kriteria 67
Gambar 5.5 Hasil Implementasi Antarmuka Hasil Rekomendasi 68
Gambar 5.6 Hasil Implementasi Antarmuka Detail Pantai 68
Gambar 6.1 Pohon Diagram Pengujian dan Analisis 69
Gambar 6.2 Data Hasil Rekomendasi 4 Kriteria 74
Gambar 6.3 Data Hasil Rekomendasi 5 Kriteria 75
Gambar 6.4 Data Hasil Rekomendasi 3 Kriteria 76

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR KODE

Kode 5.1 Kode Membuat Tabel *table*_fasilitas 51
Kode 5.2 Kode Membuat Tabel *table*_kriteria 51
Kode 5.3 Kode Membuat Tabel *table_location* 51
Kode 5.4 Kode Membuat Tabel *table*_pantai 51
Kode 5.5 Kode Membuat Tabel *table*_relasi_fasilitas 52
Kode 5.6 Kode Membuat Tabel *table*_relasi_transportasi 52
Kode 5.7 Kode Membuat Tabel *table*_transportasi 52
Kode 5.8 Kode Program Kelas MintaRekomendasiActivity 58
Kode 5.9 Kode Program Kelas HasilRekomendasiActivity 60
Kode 5.10 Kode Program Fungsi Mendapatkan Daftar Pantai 61
Kode 5.11 Kode Program Fungsi Mendapatkan Hasil Rekomendasi Pantai 64
Kode 5.12 Kode Program Fungsi Menghitung Jarak Alternatif Terhadap Solusi Ideal dan Nilai Prefensi 66

DAFTAR PERSAMAAN

Persamaan (2.1) 9
Persamaan (2.2) 9
Persamaan (2.3) 9
Persamaan (2.4) 9
Persamaan (2.5) 10
Persamaan (2.6) 10
Persamaan (2.7) 10

DAFTAR LAMPIRAN

Lampiran 1 Kasus Uji Pengujian Fungsional Menampilkan Daftar Pantai 82
Lampiran 2 Kasus Uji Pengujian Fungsional Memasukkan Bobot Kriteria 83
Lampiran 3 Kasus Uji Pengujian Fungsional Melihat Hasil Rekomendasi..... 84
Lampiran 4 Kasus Uji Pengujian Fungsional Menampilkan Informasi Pantai..... 85
Lampiran 5 Kasus Uji Pengujian Fungsional Menampilkan Peta 86

BAB 1 PENDAHULUAN

1.1 Latar belakang

Pariwisata yang berada di negara Indonesia sekarang ini dapat dikatakan efektif kontribusinya dalam menambah devisa negara. Hal itu tak terlepas dari berkembangnya kebutuhan masyarakat terhadap pariwisata. Meningkatnya kebutuhan manusia terhadap pariwisata menyebabkan sektor dinilai mempunyai prospek yang cukup menjanjikan di masa mendatang. Selain itu, sektor pariwisata juga mampu menghidupkan ekonomi masyarakat sekitarnya. Secara kumulatif dari bulan Januari hingga Agustus 2017 sebanyak 9,25 juta wisatawan mancanegara (wisman) berkunjung ke Indonesia, dimana pada periode yang sama pada tahun 2016 kunjungan wisman hanya 7,36 juta(BPS, 2018).

Lebih lanjut Menteri Pariwisata Arif Yahya mengatakan, pada tahun 2017 sector pariwisata berkontribusi 10 persen terhadap PDB nasional dan merupakan nominal tertinggi di ASEAN. Pada 2016, sector pariwisata merupakan penyumbang devisa terbesar kedua dengan jumlah USD12,5 miliar setelah CPO yang menyumbang devisa USD15,9 miliar.

Pariwisata merupakan suatu asset strategis untuk menjadi daya Tarik yang dinilai penting untuk suatu daerah dan sekaligus menjadi pendorong pembangunan pada daerah yang memiliki potensi objek wisata (Firmansyah, 2018). Salah satu daerah yang memiliki posisi yang strategis sehingga memiliki potensi objek wisata pantai yang cukup banyak adalah Malang. Keindahan pantai yang alami dan masih banyak belum tersentuh tangan manusia menjadi salah satu keunggulan pantai yang berada disekitar Malang ini.

Wisata pantai yang terdapat di daerah Malang cukup banyak menyebabkan pengunjung bingung untuk memilih pantai yang ingin dikunjungi, hal itu juga menyebabkan masih banyaknya pantai-pantai yang masih sepi wisatawan. Selain karena jumlahnya yang banyak, hal ini bisa disebabkan juga karena kurangnya informasi yang dimiliki oleh wisatawan. Padahal setiap pantai di Malang memiliki keunggulan dan ciri khas masing-masing.

Sekarang kepemilikan *smartphone* tidak hanya didasari pada fungsi utamanya saja yaitu alat komunikasi, akan tetapi fitur-fitur yang ditawarkan serta desain produk juga menjadikannya dasar dalam memepertimbangkan unttuk memilih jenis atau merk *smartphone* yang akan digunakan. Seperti yang kita ketahui, bahwa perkembangan teknologi semakin hari semakin pesat dan sekarang kita mengenal istilah *smartphone* yang memiliki banyak fitur dan kecanggihan yang hampir menyamai sebuah computer. Bahkan smartphone saat ini telah mengubah gaya hidup masyarakat, mulai dari melihat peta, memesan makanan, ngobrol dengan teman, membeli tiker, mentransfer uang, update status social media,

bermain game, bahkan sudah banyak yang melakukan edit video di smartphonenya.

Berdasarkan masalah yang sering terjadi pada wisatawan dan banyaknya kepemilikin smartphone pada masyarakat, maka sebuah sistem yang dapat memberikan pilihan rekomendasi pantai yang harus dikunjungi, beserta informasi terkait lokasi, dan rute menuju lokasi menjadi sesuatu yang dubutuhkan saat ini. Sistem akan memberikan pilihan rekomendasi wisata pantai berdasarkan beberapa parameter, yaitu: harga, jarak dari lokasi user dan fasilitas.

Terdapat beberapa metode yang bisa digunakan pada sistem rekomendasi ini, yaitu metode *Technique For Order Preference by Similiarity to Ideal Solution* (TOPSIS), *Analytic Hierarchy Process* (AHP), Weighted Product, *Simple Additive Weighting* (SAW) dan metode lainnya. Dari beberapa metode yang bisa digunakan untuk sistem rekomendasi, penulis memilih metode TOPSIS dengan alasan karena TOPSIS dapat melakukan perangkingan alternatif, dimana alternatif yang terpilih mempunyai jarak paling dekat dengan solusi ideal positif dan jarak paling jauh dengan solusi ideal negatif (Gunawan, 2014). Bukan hanya itu, metode TOPSIS juga dipilih karena konsepnya yang tidak kompleks, mudah untuk dimengerti, efisien dalam komputasi, dan juga dapat menghitung nilai kinerja dari alternatif keputusan kedalam bentuk matematis tidak kompleks(Maharani, 2014).

Selain karena beberapa alasan metode TOPSIS dipilih diatas. Metode TOPSIS sudah sering dipakai dalam penelitian para peneliti. Salah satu penelitian yang pernah menggunakan metode TOPSIS adalah penelitian yang pernah dibuat Bustanul Arifin tahun 2015, yang berjudul "Sistem pendukung keputusan rekomendasi pemilihan mobil bekas dengan menggunakan metode TOPSIS". Pada penilitian tersebut bukan hal yang tidak mungkin untuk mengimplementasikan metode TOPSIS pada kasus pemilihan mobil bekas. Selain itu rekomendasi dari penilitian tersebut adalah bagaimana mengimplementasikannya pada aplikasi mobile Android (Bustanul Arifin, 2015).

Penelitian selanjutnya yang menggunakan metode TOPSIS adalah penelitian yang dilakukan oleh Reza Perdana dengan judul "Sistem Rekomendasi Tempat Wisata Kuliner Menggunakan Metode TOPSIS Berbasis Mobile" pada tahun 2014. Dimana pada penilitian ini lebih berfokus pada bagaimana aplikasi mampu menyampaikan rekomendasi tempat wisata kuliner yang halal dan sesuai dengan keinginan pengguna. Hasil dari penelitian tersebut juga menyatakan bahwa pengguna yang merasakan manfaat aplikasi mencapai angka 82% dan pengguna yang mendapatkan informasi serta pengalaman baru dalam penggunaan aplikasi mencapai angka hingga 96%.(Reza Perdana, 2014).

Selanjutnya ada juga penelitian yang dibuat oleh Amanda Ayu Larasati yang berjudul "*Development Decision Support System of Choosing Medicine using TOPSIS Method"* pada tahun 2017. Pada penilitian tersebut peneliti menggunakan metode TOPSIS untuk mendapatkan rekomendasi terkait obat yang akan dipilih oleh pengguna. Berdasarkan penelitian tersebut dapat diambil kesimpulan bahwa

metode TOPSIS dapat diaplikasikan pada system rekomendasi untuk memilih obat. (Larasati, Setianingrum and Wardhani, 2017)

Berdasarkan uraian diatas, adanya suatu sistem yang dapat memberikan rekomendasi wisata pantai berbasis lokasi dengan menerapkan metode TOPSIS ini diharapkan dapat mempermudah promosi dan sosialisasi objek wisata pantai yang ada di daerah sekitar Malang. Oleh karena itu, penulis memilih judul : "**IMPLEMENTASI TOPSIS PADA SISTEM REKOMENDASI TEMPAR WISATA PANTAI DI SEKITAR MALANG BERBASIS MOBILE ANDROID"**.

1.2 Rumusan masalah

Seperti yang sudah dijabarkan pada latar belakang, dan juga sesuai denga tujuan dibuatnya penelitian ini, maka terdapat 3 rumusan masalah skripsi, yaitu:

1. Bagaimana membangun sistem yang dapat memberikan rekomendasi tempat wisata pantai di sekitar Malang dengan berbasiskan lokasi?
2. Bagaimana sistem rekomendasi tempat wisata pantai di sekitar Malang dapat terintegrasi dengan metode TOPSIS?
3. Bagaimana pengujian sistem rekomendasi tempat wisata pantai di sekitar Malang menggunakan metodeTOPSIS?

1.3 Tujuan

1. Membangun sistem yang dapat memberikan rekomendasi pantai di sekitar Malang berbasis lokasi.
2. Mampu mengintegrasikan metode TOPSIS untuk mendukung rancangan sistem rekomendasi tempat wisata pantai di sekitar Malang.
3. Dapat mengetahui hasil pengujian sistem.

1.4 Manfaat

Dengan penelitian ini, terdapat manfaat yang bisa didapatkan, yaitu:

- Dengan dirancangnya aplikasi ini, wisatawan dapat lebih mudah dalam menentukan objek wisata pantai yang akan dikunjungi.
- Membantu wisatawan untuk mendapatkan informasi terkait wisata pantai Malang

1.5 Batasan masalah

Dengan luasnya cakupan pembahasan mengenai system rekomendasi tempat wisata pantai dan agar penelitian ini dapat fokus dan menghindari semakin melebarnya masalah, maka akan dilakukan pembatasan pembahasan dengan tujuan penelitian dapat terfokus pada masalah yang yang diangkat. Lingkup pembahasan antara lain:

1. Data objek wisata pantai yang digunakan hanya di sekitar Kota Malang.

2. Kriteria yang digunakan untuk merekomendasikan wisata pantai adalah harga tiket masuk, *rating,* jarak dan fasilitas.
3. Metode TOPSIS merupakan metode yang akan digunakan pada sistem.
4. GPS dan Internet dianggap bekerja dengan baik dan dapat secara langsung digunakan pada aplikasi. Tidak diperlukan pembahasan untuk perancangan dan implementasi terkait GPS dan internet.

1.6 Sistematika pembahasan

Pada bagian ini terdapat penjelasan singkat terkait struktur skripsi dari pendahuluan hingga penutup. Diharapkan bagian ini dapat membantu untuk memahami isi dalam skripsi ini. Pembahasan terkait susunan skripsi terbagi kedalam 7 (tujuh) bab, yaitu:

BAB I : PENDAHULUAN

Menjelaskan latar belakang masalah, rumusan masalah, manfaat dari diadakannya penelitian, tujuan, batasan masalah atau ruang lingkup pembahasan, dan sistematika penulisan dari sistem rekomendasi tempat wisata pantai di sekitar Malang.

BAB II : LANDASAN KEPUSTAKAAN

Menjelaskan secara singkat terkait penelitian yang sudah pernal dilakukan sebelumnya dan juga materi yang digunakan sebagai dasar dalam pembuatan sistem rekomendasi tempat wisata pantai di sekitar Malang.

BAB III : METODOLOGI

Menjelaskan tentang tahapan metodologi yang akan dilakukan dalam penelitian skripsi pembangunan sistem rekomendasi tempat wisata pantai di sekitar Malang.

BAB IV : ANALISIS DAN PERANCANGAN

Menjelaskan tentang gambaran umum sistem, identifikasi aktor dan kebutuhan fungsional. Selain itu menjelaskan juga terkait perancangan dari analisis kebutuhan yang ada.

BAB V : IMPLEMENTASI

Menjelaskan tentang implementasi sistem berdasarkan pada perancangan sistem.

BAB VI : PENGUJIAN DAN ANALISIS

Menjelaskan terkait pengujian serta analisis terkait sistem yang sudah dibuat.

BAB VII : PENUTUP

Menjelaskan terkait kesimpulan yang telah dilakukan dan juga masukan saran untuk pengembangan kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Penelitian Terdahulu

Dasar teori atau temuan hasil penelitian sebelumnya dapat dijadikan sebagai data pendukung. Pada penelitian ini, penilitian terdahulu yang akan dijadikan sebagai acuan ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Penelitian Rujukan

No	Peneliti	Judul	Hasil
1	Reza Perdana, 2014.	Sistem Rekomendasi Tempat Wisata Kuliner Menggunakan Metode TOPSIS Berbasis Mobile.	Aplikasi mampu menyampaikan rekomendasi tempat wisata kuliner yang halal dan sesui dengan keinginan pengguna. Hasil dari penelitian juga menyatakan bahwa 82% pengguna merasakan manfaat dari aplikasi dan 96% mendapatkan informasi dan pengalaman menggunakan aplikasi baru
2	Bustanul Arifin, 2015.	Sistem Pendukung Keputusan Rekomendasi Pemilihan Mobil Bekas Dengan Menggunakan Metode TOPSIS.	Metode TOPSIS dapat diimplementasikan pada kasus pemilihan mobil bekas. Selain itu rekomendasi dari penulis untuk penelitian selanjutnya adalah bagaimana mengimplementasikannya pada aplikasi mobile Android.

3	Larasati, Setianingrum, dan Wardhani, 2017.	*Development Decision Support System of Choosing Medicine using TOPSIS Method.*	Metode TOPSIS dapat diimplementasikan pada kasus pemilihan obat kesehatan.

2.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan pertama kali diperkenalkan oleh Michael S. Scott Morton pada tahun 1970, kala itu dikenal dengan nama Management Decision Model (Sprague, 1982). Dasar dari sistem pendukung keputusan adalah karakterisasi yang dilakukan oleh sistem komputer interaktif yang membantu pengambilan keputusan menggunakan bantuan data serta model dalam mengatasi masalah-masalah yang tidak terstruktur. Sistem ini di desain agar membantu dalam mengambil keputusan dimulai dengan mengindentifikasi permasalahan, mengambil data-data yang relevan, melakukan pendekatan untuk mengambil keputusan dan melakukan evaluasi terhadap pilihan secara interaktif. (Ginting et al., 2017)

Berdasarkan pernyataan Raymond McLeod, Jr (1998) definisi dari sistem pendukung keputusan adalah system yang memiliki tujuan unutk membantu daam proses manajemen penyelesaian masalah (McLeod, 1998). Selain itu ada juga pendapat yang mengatakan bahwa sistem pendukung keputusan adalah suatu sistem komputer yang memberikan alternative pilihan agar dapat membantu mengatasi manajemen masalah menggunakan data dan model. Dengan begitu dapat ditarik kesimpulan bahwa sistem pendukung keputusan merupakan sistem informasi yang dapat memberikan bantuan terhadap para manajer dalam memberikan alternative pilihan dalam menyelesaikan masalah yang dihadapi. (Ginting et al., 2017)

2.2.1 Faktor Utama Sistem Pendukung Keputusan

Terdapat 4 komponen utama dalam system pengukung keputusan yang saling berhubungan satu sama lain. Komponen sistem pendukung keputusan tersebut terdiri dari data *management, model management, communication,* dan *knowledge management* (Risza, 2016).

1. Data *Management*

   Database yang mengandung data-data lengkap dan relevan terhadap semua situasi dan juga dikendalikan oleh *software* yang disebut dengan *Database Management System* (DBMS) merupakan salah satu yang termasuk *data management.*

2. *Model Management*

Di dalamnya terdapat model statistical, finansial, *management science,* dan juga semua model kuantitatif, hal itu menyebabkan kemampuan untuk memberikan nilai kepada sistem untuk melakukan analisis dan manajemen.

3. *Communication*

   Merupakan antarmuka untuk interaksi sistem dengan pengguna. Pengguna dapat berkomunikasi dengan memberikan *input* terhadap sistem melalui bagian ini.

4. *Knowledge Management*

   Merupakan subsistem yang bersifat *optional*. Subsistem ini dapat bertindang sebagai pendukung subsistem yang lain ataupun bertindak secara mandiri.

2.3 Metode *Technique For Order Preference by Similarity to Ideal Solution* (TOPSIS)

Technique For Order Preference by Similarity to Ideal Solution atau biasa disingkat TOPSIS merupakan metode yang pertama kali diperkenalkan pada tahun 1981. Metode ini memberikan hipotesis terkait dua alternative buatan yaitu alternative untuk solusi optimal dan alternative untuk solusi tidak optimal. Dasar dari pemikiran ini berasal dari konsep kompromi dimana solusi alternative yang memiliki jarak paling dekat terhadap solusi optimal dan memiliki jarak yang paling jauh dari solusi tidak optimal (Lokare dan Jadhav, 2016).

2.3.1 Langkah-langkah TOPSIS

Langkah yang digunakan dalam mencari solusi dengan metode TOPSIS pada umumnya dikerjakan dengan melewati tahapan berikut (Perdana, 2014):

1. Memasukkan nilai bobot kriteria dan data alternatif

   Merupakan suatu tahapan yang biasa dikenal dengan istilah pra-proses. Dimana pada tahap ini dimasukkan data alternative ke dalam system dan kriteria-kriteria yang mempengaruhi pengambilan suatu keputusan diberi suatu nilai bobot berdasarkan presentasi kriteria tersebut dalam menentukan pengambilan keputusan. Segala macam cara maupun metode yang absah dapat dilakukan untuk mentukan bobot kriteria.

2. Menentukan matriks keputusan ternormalisasi

   Berikut merupakan langkah yang digunakan untuk mendapatkan nilai normalisasi:

   a. Menghitung nilai kuadrat tiap kriteria

   b. Lakukan penjumlahan nilai kuadrat krteria

c. Lakukan perhitungan untuk mencari nilai normalisasi dari setiap elemen untuk tiap kriteria.

Elemen r_{ij} hasil normalisasi ditunjukkan oleh persamaan berikut:

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

penjelasan:

r_{ij} = nilai normalisasi *decision matrix* R;

i = 1,2,3,...,m;

j = 1,2,3,...m;

3. Menentukan matriks keputusan yang ternormalisasi terbobot

Jika nilai W=(w1, w2,……., wn), untuk mencari nilai normalisasi terbobot matriks V ditunjukkan oleh persamaan berikut:

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

Setelah dilakukan pembobotan, maka matriks keputusan V_{ij} dapat dijalankan sebagai matriks keputusan yang etlah dinormalisasi dan diberi bobot.

4. Menentukan matriks ideal positif dan matriks ideal negatif

Dengan solusi ideal positif dinotasikan sebagai $A^{+,}$ dan solusi ideal negative dinotasikan sebagai A-. Maka persamaan untuk menentukan solusi ideal negative dan solusi ideal positif ditunjukkan oleh persamaan berikut:

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

$$A^+ = \left\{\left(\max v_{ij} \middle| j \in J\right) \quad \left(\min v_{ij} \middle| j \in J`\right), i = 1,2,3,...,m\right\} = \left\{v_1{}^-, v_2{}^-, ..., v_m{}^-\right\} \quad 2.4)$$

dimana:

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

J = {j = 1,2,3,...,m dan j berhubungan dengan *benefit* kriteria};

J' = {j = 1,2,3,...,m dan j berhubungan dengan *cost* kriteria};

5. Menghitung jarak setiap matriks alternatif dengan matriks solusi ideal negatif dan positif

   Dimana S_i^+ merupakan jarak (dalam pandangan *Euclidean*) antara alternative solusi dengan solusi ideal positif/optimal. Sedangkan, S_i^- merupakan jarak (dalam pandangan *Euclidean*) antara alternative solusi dengan solusi ideal negative/tidak optimal. Maka persamaan untuk menghitung separasi ditunjukkan oleh persamaan berikut:

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

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

6. Menghitung nilai prefensi

   Nilai prefensi merupakan kedekatan relative terhadap solusi optimal atau biasa dinotasikan C_i^+ merupakan suatu perbandingan antara jarak alternative solusi optimal S_i^+ dengan jarak alternative solusi tidak optimal S_i^-. Untuk mencari kedekatan relative ditunjukkan oleh persamaan berikut:

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

7. Perangkingan hasil akhir dengan menggunakan nilai prefensi

   Merupakan proses untuk memberikan rangking pada alternatif solusi dimana urutan ranking dapat diurut berdasarkan pada urutan C_i^+ yang terbesar hingga yang terkecil. Maka dari itu, alternative pilihan terbaik merupakan alternative pilihan yang memiliki jarak paling dekat terhadap solusi ideal positif dan memiliki jarak paling jauh terhadap solusi ideal negatif. Dimana alternatif pilihan dengan nilai C_i^+ terbesar merupakan solusi terbaik.

2.4 Android

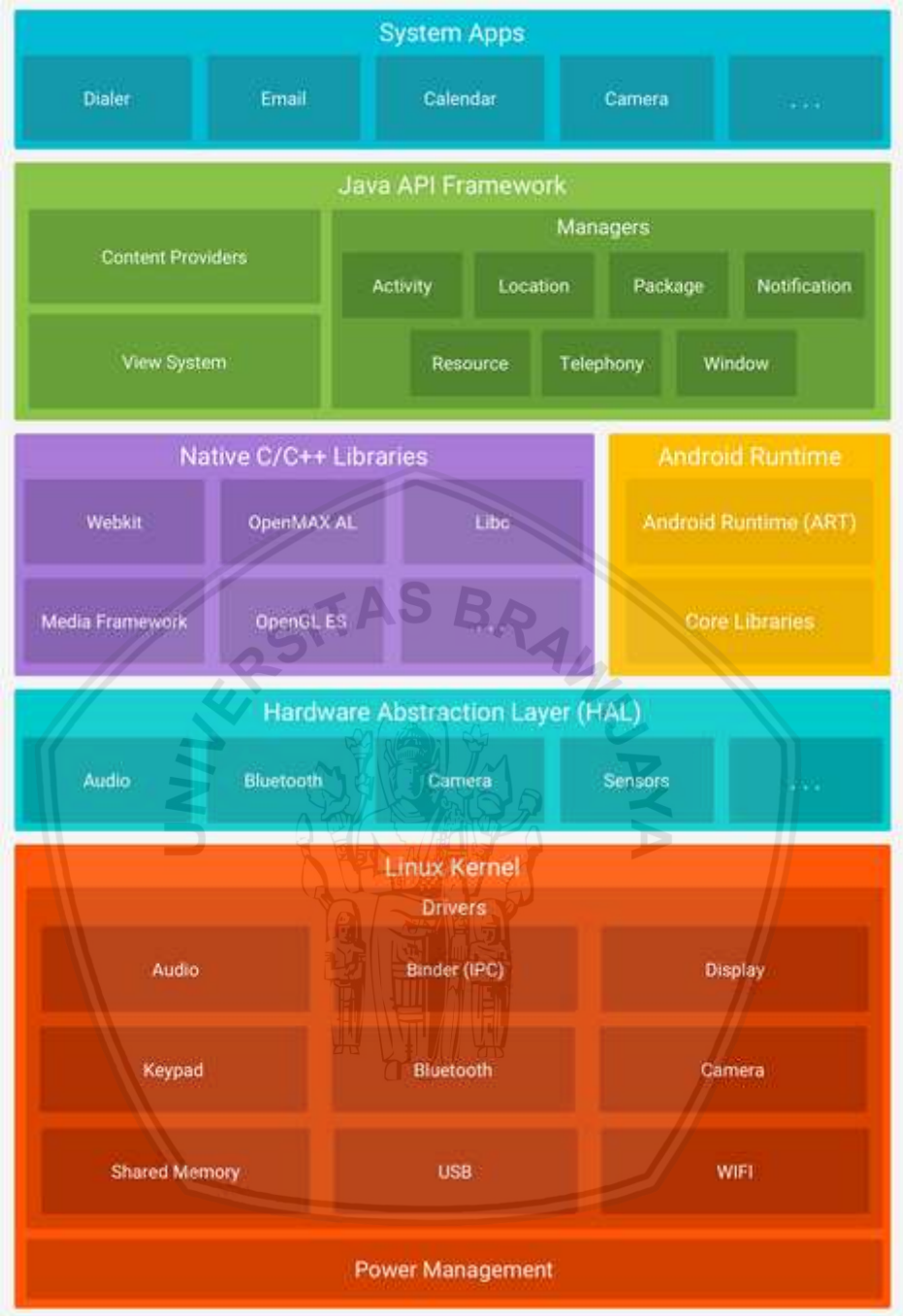
Merupakan suatu kumpulan *sofware* yang berada pada *smartphone* yang didalamnya terdapat *operating system*, *middleware* dan aplikasi(Safaat, 2014). Agar dapat membantu pengembang dalam membuat suatu aplikasi, Android mengizinkan akses yang luas dalam penggunaan *library* dan *tools* untuk bisa digunakan agar meningkatkan kualitas aplikasi(Developers, 2018).

Google Inc. Selaku perusahaan yang telah membuat Android, menjadikan Android sebagai sistem yang memiliki sifat *open source,* dimana hal itu

menyebabkan para pengembang dapat mengeksploitasi Android secara lebih leluasa. Android *Software Development Kit* (SDK) merupakan tempat dikumpulkannya alat dan API atau biasa disebut *Application Programming Interface* yang berguna dalam membuat dan mengembangkan aplikasi dalam *platform* Android (Developers, 2018).

2.4.1 Arsitekrut Android

Gambaran umum terkait arsitertur android dapat dilihat pada gambar 2.1



Gambar 2.1 Arsitektur Android

Berikut ini merupakan penjelasan dari arsitektur Android tersebut .

1. **Linux Kernel**

   Pada bagian paling bawah dari Android adalah Linux Kernel. Android menggunakan modifikasi Linux Kernel versi 2.6. Bagian ini bertanggung jawab untuk mengelola dan berkomunikasi dengan perangkat keras di mana Android

berjalan. Pemilihan Linux Kernel sebagai inti dari Android adalah karena dukungan dan kestabilan terhadap berbagai macam komponen perangkat keras. Pada bagian ini disediakan driver (program pengendali) perangkat keras, pengelola memori, pengelola proses, pengelola jaringan dan keamanan (Developers, 2018).

2. **Hardware Abstraction Layer (HAL)**

   Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL tediri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut (Developers, 2018).

3. **Android Runtime**

   Pada bagian inilah aplikasi Android berjalan. Inilah bagian middleware Android. Aplikasi Android ditulis menggunakan bahasa pemrograman Java dan untuk menjalankannya diperlukan sebuah virtual machine yang akan menerjemahkan instruksi-instruksi program Java tersebut ke instruksi yang dimengerti oleh sistem operasi. Namun berbeda dengan platform lainnya, Android tidak menggunakan Java Virtual Machine (JVM), melainkan menggunakan Dalvik Virtual Machine (DVM). Dalvik Virtual Machine adalah sebuah virtual machine yang dioptimasi untuk perangkat yang memiliki memori kecil, sumber tenaga terbata dan kemampuan prosesor yang kecil (Developers, 2018).

4. **Libraries**

   Pada bagian ini terdapat kumpulan *code* yang bisa digunakan pada komponen atau program lain. Setiap komponen pada *libraries* memiliki fungsinya masing-masing, yakni: (Developers, 2018)

   1. *Surface manager:* untuk manajemen tampilan gambar serta komposisi komponen pada layar.
   2. *Open GL|ES: engine* grafik 3 dimensi.
   3. SGL: *engine* grafik 2 dimensi.
   4. Media Framework: *codecs* yang digunakan untuk melakukan perekaman dan pemutaran format video dan audio.
   5. Free rype: berfungsi melakukan rendering pada *font*.
   6. SSL: untuk proses pengamanan komunikasi jaringan.
   7. SQLite: untuk mengelola database yang digunakan untuk proses penyimpanan data.

8. Webkit: browser engine.
9. Libc: *library* C standar.

5. **Application Framework**

   Merupakan framework yang dapat digunakan oleh pengembbang untuk membuat aplikasi Android. Bagian ini juga mencantumkan program yang mengelola fungsi dasar seperti informasi lokasi, fungsi telepon, pengelolaan perpindahan antarprogram, dan sebagainya. Pengembang memiliki akses framework yang sama dengan aplikasi-aplikasi utama di Android. (Developers, 2018).

6. **Applications**

   Pada bagian ini terdapat aplikasi-aplikasi yang dapat digunakan oleh pengguna perangkat Android. Selain aplikasi inti seperti yang ada di diagram arsitektur, aplikasi-aplikasi tambahan yang dipasang sendiri oleh pengguna akan menempati bagian ini dan memiliki hak akses yang sama terhadap Application Framework. Apabila diperlukan, pengguna ataupun vendor dapt menggantikan aplikasi inti yang ada dengan aplikasi buatan sendiri yang memiliki fungsi yang sama. Hal ini membuat kostumisasi sistem operasi Android jauh lebih bebas dibandinkan perangkat mobile lainnya (Developers, 2018).

2.4.2 *Android Software Development Kit* (SDK)

Android SDK merupakan tools *Application Programming Interface* atau biasa disingkat API yang diperlukan untuk memulai pengembangan aplikasi Android. Sebagai suatu platform aplikasi, Android memberikan kesempatan kepada para pengembang untuk membuat aplikasi sesuai dengan kebutuhan(Safaat, 2015).

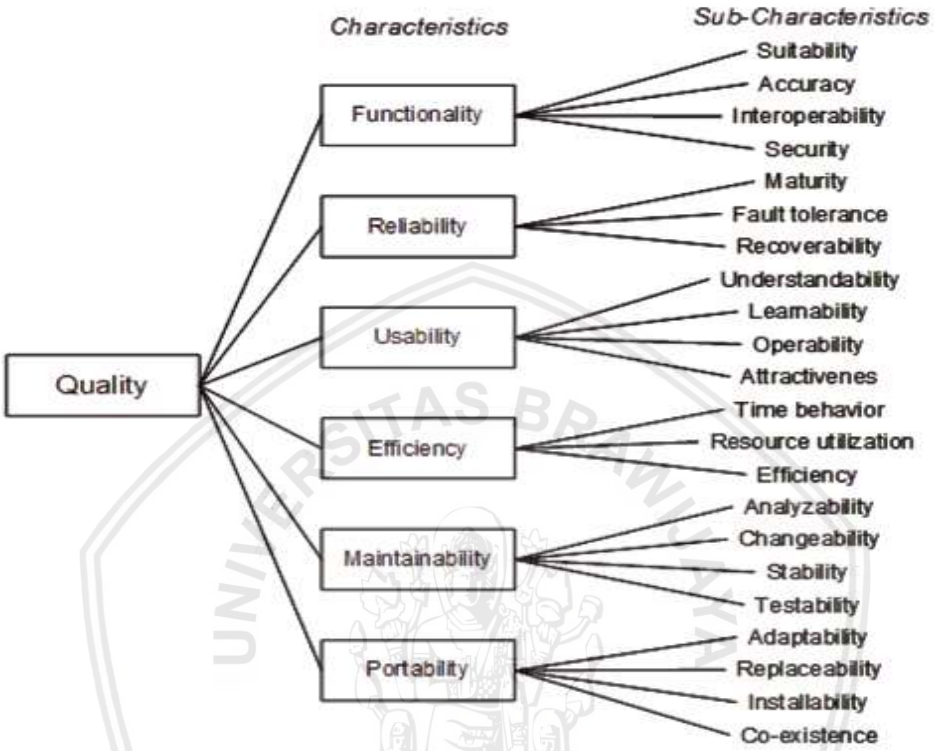
2.5 Pengujian Perangkat Lunak

Pengujian merupakan suatu bentuk tindakan investigasi terhadap perangkat lunak yang sedang dikembangkan agar diketahui kualitas dari perangkat lunak. Melalui pengujian pula diharapkan pengembang aplikasi dapat mengetahui apakah terdapat kesalahan pada perangkat lunak yang sedang dikembangkan. Selain itu, melalui pengujian perangkat lunak pula hasil catatan dari pengujian dapat di evaluasi untuk pengembangan di masa depan.

2.5.1 ISO 9126

ISO 9126 merupakan suatu standar yang penting dalam bidang penjamin kualitas. Dari model tersebut, kualitas suatu perangkat lunak dinilai dan

diklasifikasikan dalam struktur hierarki dari karakteristik dan sub karasteristik. Masing-masing karakteristik ini kemudian dibagi-bagi lagi menjadi 21 sub-karakteristik. Karakteristik-karakteristik yang didefinisikan dapat diaplikasikan dalam setiap perangkat lunak apapun. Gambar 2.2 menunjukkan struktur hierarki dari ISO 9126.



Gambar 2.2 Struktur hierarki ISO 9126

2.5.2 Pengujian Black Box

Pengujian *Black Box* adalah sauatu metode uji yang lebih fokus pada spesifikasi fungsional. Penguji bisa memberikan definisi kondisi masukan dan dilakukan pengetesan program (Ach Fauzan, 2017). Pada pengujian *Black Box,* kebenaran fungsionalitas dari perangkat lunak hanya dapat dilihat dari hasil keluaran perangkat lunak yang diuji tanpa memperhatikan baigaimana proses yang terjadi dibelakangnya. Berdasarkan hasil keluaran perangkat lunak tersebutlah, penguji dapat mengetahui dan mengukur kemampuan dari perangkat lunak juga mengetahu kesalahan-kesalahan dari perangkat lunak.

a) Kelebihan Pengujian *Black Box*
    1. Memilih subset test dengan efisien dan efektif.
    2. Menemukan adanya kekurangan pada perangkat lunak
    3. Memaksimalkan testing investmen pada proses pengujian

b) Kelemahan pengujian *Black Box*

1. Penguji tidak akan pernah meyakini perangkat lunak tersebut sudah lulus uji atau tidak.

2.5.3 Pengujian *Rank Consistency*

Merupakan suati pengujian yang dilakukan agar dapat mengetahui ketahanan hasil rekomendasi tempat wisata pantai terhadap perubahan data. Dari pengujian ini kita dapat mengetahui seberapa konsisten data hasil rekomendasi. Pengujian ini bisa dilakukan dengan melakukan penambahan maupun pegurangan data alternatif atau kriteria. Pengujian ini memiliki tujuan untuk mengetahui apakah suatu sistem dapat bertahan dengan perubahan data (Rodrigues et al., 2014).

BAB 3 METODOLOGI

Pada bab ini akan menjabarkan terkait tahapan-tahapan metodologi yang ada pada penelitian skripsi. Gambar 3.1 menjelaskan terkait alur metodologi yang digunakan.

Gambar 3.1 Alur Metodologi

3.1 Studi Literatur

Merupakan tahap dimana penulis belajar serta melakukan pengumpulan data penelitian yang mirip dan sudah dilakukan dari berbagai literatur yang dapat bersumber dari buku, jurnal, *paper* maupun internet. Dasar teori merupakan penjelasan terkait dasar teori yang digunakan.

3.2 Pengumpulan Data

Pengumpulan data merupakan tahapan penulis mencari data dan mengumpulkan data terkait tempat wisata pantai di Malang yang dibutuhkan sebagai bahan untuk penelitian. Data tempat wisata pantai yang dibutuhkan antara lain adalah nama lokasi pantai, alamat, koordinat lokasi (*latitude* dan

longitude), harga tiket masuk, fasilitas dan rating lokasi. Data tempat wisata pantai sebagai bahan penelitian diperoleh dari penelitian sebelumnya dengan topik judul "*Pengembangan Pemanfaatan Peta Digital Untuk Mendukung Sistem Informasi Pariwisata Pantai Di Kabupaten Malang*" (Noor, 2012)

3.3 Analisis Kebutuhan

Merupakan langkah awal untuk menentukan kebutuhan perangkat yang akan dibuat dalam penelitian. Analisis kebutuhan sistem rekomendasi tempat wisata pantai terbagi menjadi lima, yakni : deskripsi umum, pengidentifikasian aktor, kebutuhan fungsional, *usecase* dan *usecase scenario.*

3.4 Perancangan Sistem

Perancangan sistem yang dibuat merupakan perancangan yang mengacu pada analisis kebutuhan yang sudah dilakukan sebelumnya. Perancangan didasarkan pada *Object Oriented Design* dimana mengacu pada *Unified Modelling Languange* (UML). Pada penelitian penulis menggunakan 4 pemodelan antara lain: perancangan sistem dimulai dengan melakukan pemodelan rancangan activity diagram, sequence diagram dan class diagram. Kemudian pada tahap akhir dilanjutkan dengan melakukan pemodelan basis data, dan perancangan navigasi antarmuka.

3.5 Implementasi Sistem

Sistem yang dibuak merupakan implementasi yang mengacu pada perancangan yang sudah dibuat sebelumnya. Implementasi sistem diawali dengan penjabaran lingkungan sistem, batasan, implementasi class, basis data menggunakan MySQL, kode program dan antar muka.

3.6 Pengujian Sistem

Pengujian sistem merupakan langkah yang dilakukan dengan tujuan dapat mengetahui bahwa aplikasi dapat bekerja sesuai yang diharapkan. Pengujian sistem terdiri dari 2 sub bab, yaitu: pengujian dan analisis. Dimana pengujian terdiri dari pengujian *black box*, pengujian validasi algoritma dan pengujian *rank consistency*. Begitupun analisis terdiri dari analisis pengujian *black box,* analisis pengujian validasi algoritma dan analisis *rank consistency*.

3.6.1 Pengujian Black Box

Pada pengujian ini lebih fokus pada kebutuhan fungsionalitas perangkat. Pengujian *black box* mencoba untuk dapat mengetahui kesalahan pada fungsional sistem.

3.6.2 Pengujian Validasi Algoritma

Pada pengujian ini dilakukan perbandingan antara hasil rekomendasi dari keluaran sistem dengan perhitungan manual.

3.6.3 Pengujian *Rank Consistency*

Pada pengujian *rank consistency* dilakukan penambahan dan pengurangan kriteria pada hasil rekomendasi.

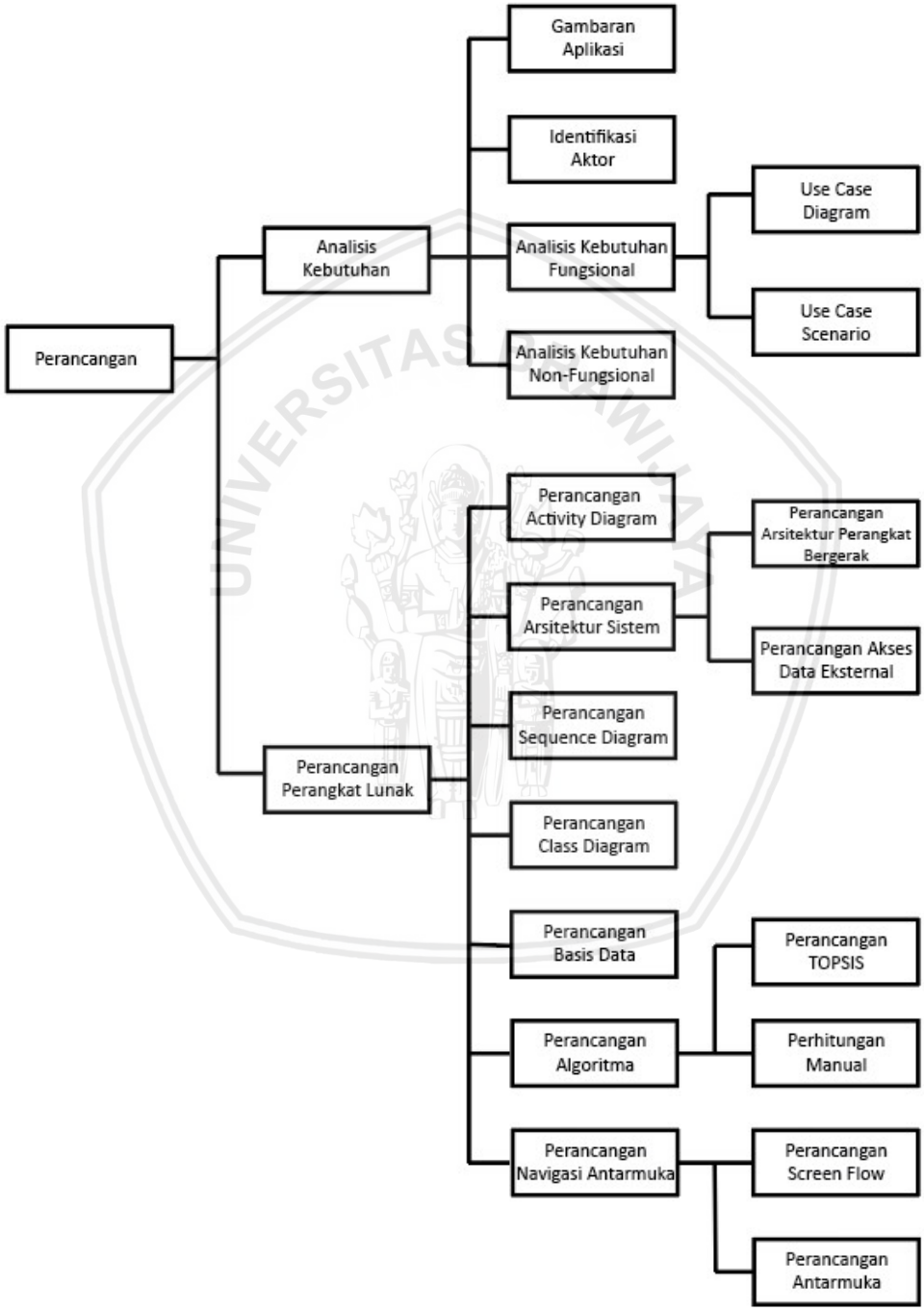
3.7 Pengambilan Kesimpulan dan Saran

Setelah semua tahap dalam penelitian implementasi metode TOPSIS pada sistem rekomendasi tempat wisata pantai di sekitar Malang telah dilakukan maka dapat ditarik suatu pengambilan kesimpulan. Kesimpulan dapat diambil dari mulai analisis dan perancangan hingga hasil pengujian sistem. Terakhir adalah saran untuk penelitian di masa depan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 4 ANALISIS DAN PERANCANGAN

Bab ini akan menjelaskan terkait perancangan dari system yang akan dibuat. Penjelasasn terkait struktur dan langkah analisis serta perancangan pada sistem rekomendasi objek wisata pantai di Malang menggunakan metode TOPSIS terdapat pada Gambar 4.1



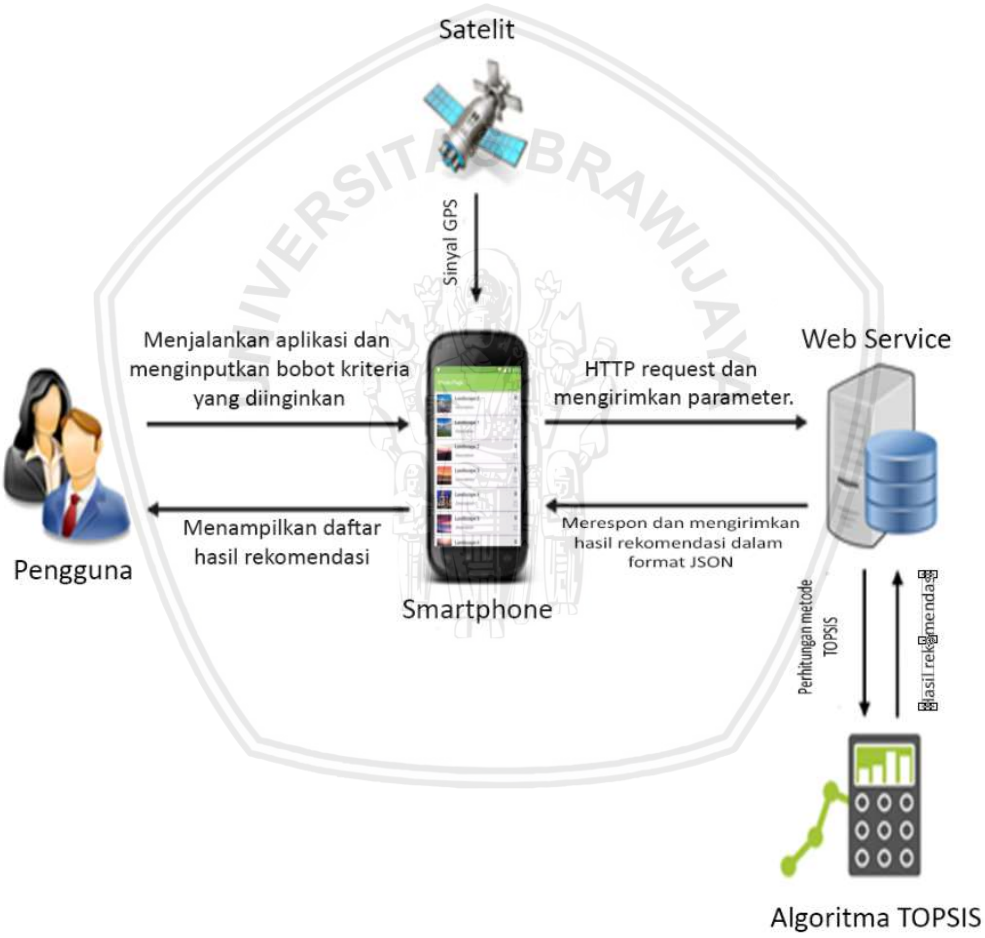
Gambar 4.1 Diagram Pohon Perancangan

4.1 Analisis Kebutuhan

Identifikasi kebutuhan terhadap system dilakuakan pada tahap ini. Beberapa tahapan dilakukan, antara lain: gambaran sistem secara umum, mengidentifikasi aktor, kebutuhan fungsional, dan kebutuhan non fungsional. Memberikan gambaran secara jelas kepada pengguna merupakan tujuan pada tahap ini.

4.1.1 Gambaran Aplikasi

Memberikan gambaran terkait desain arsitektur dari system yang akan dirancang. Tahap ini merupakan tahapan awal dari perancangan sistem. Gambaran aplikasi secara umum ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Gambaran Umum

Pada saat pengguna membuka aplikasi melalui perangkat bergerak Android yang dimilikinya. Aplikasi menyaratkan fitur paket data atau *Wi-Fi* dan sensor GPS di perangkat bergerak dalam keadaan menyala (ON) agar aplikasi dapat mengakses data ke *web service* dan mendeteksi koordinat lokasi perangkat bergerak pengguna. Setelah membuka aplikasi, akan muncul menu minta rekomendasi pantai. Pada menu minta rekomendasi pantai terdapat fitur untuk

mengisi *input* bobot kriteria yang diinginkan oleh pengguna. Kriteria yang ada pada menu minta rekomendasi pantai berupa *rating,* harga tiket masuk, jarak tempuh, fasilitas dan transportasi.

Data dari perangkat bergerak berupa *rating,* harga tiket masuk, fasilitas, transportasi dan koordinat perangkat bergerak (berupa *latitude* dan *longitude*) ini nantinya akan dikirim ke *web service* menggunakan alamat URL. Setelah diterima oleh *web service*, data ini akan diproses dan dihitung menggunakan algoritma TOPSIS untuk mendapatkan hasil rekomendasi pantai yang sesuai dengan input pengguna. Hasil perhitungan akan dikirim kembali ke perangkat bergerak pengguna dalam format JSON dan akan ditampilkan pada perangkat bergerak berupa daftar rekomendasi pantai yang sesuai dengan kriteria yang direkomendasikan oleh system.

4.1.2 Identifikasi Aktor

Melakukan identifikasi aktor yang berhubungan dengan system merupakan tujuan dari tahapan ini. Deskripsi terkait aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna Perangkat Bergerak	Pengguna dapat menginput bobot *rating,* harga tiket masuk, jarak tempuh, fasilitas dan transportasi. Pengguna akan mendapatkan daftar hasil rekomendasi pantai dan dapat melihat peta digital via google maps dari tempat wisata yang dipilih.

4.1.3 Analisis Kebutuhan Fungsional

Spesifikasi kebutuhan fungsional dapat dilihat pada Tabel 4.2.

Tabel 4.2 Spesifikasi Kebutuhan Fungsional

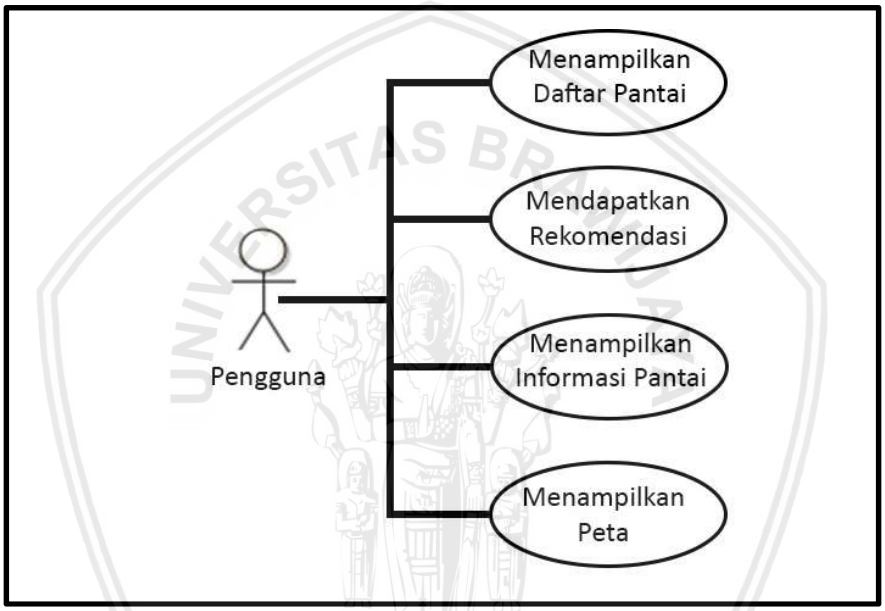
Identifier	Kebutuhan Fungsional	Nama *Use Case*
SKF-01	Sistem dapat memberikan fitur untuk menampilkan daftar pantai yang berada pada *database*.	Menampilkan daftar pantai
SKF-02	Sistem harus menyediakan fitur untuk mendapatkan rekomendasi tempat wisata pantai.	Mendapatkan Rekomendasi
SKF-03	Sistem dapat menampilkan informasi terkait tempat wisata pantai yang dipilih	Menampilkan informasi pantai
SKF-04	Sistem dapat menampilkan peta digital dan rute yang bisa dilalui	Menampilkan peta

	untuk menuju tempat wisata yang dipilih.	

Selanjutnya spesifikasi kebutuhan fungsional akan dijabarkan lebih lanjut dengan *usecase diagram* dan skenario *usecase.*

4.1.4 Use Case Diagram

Memberikan gambaran yang jelas terkait kebutuhan system dalam bentuk diagram merupakan penjelasan dari *usecase diagram*. Kebutuhan fungsionallah yang menjadi dasar dibuatnya *usecase diagram*. Gambar 4.3 menunjukkan *usecase diagram* untuk aktor pengguna.



Gambar 4.3 *Use Case Diagram* Kebutuhan Fungsional

1. Spesifikasi Use Case

Menjelaskan mengenai detail *usecase* menampilkan daftar pantai (SKF-01), mendaoatkan rekomendasi (SKF-02), menampilkan detail informasi pantai (SKF-03), dan menampilkan peta (SKF-04) yang terdapat pada *usecase diagram.* Penjelasakan spesifikasi *use case* ditunjukkan pada Tabel 4.3 sampai Tabel 4.6.

Tabel 4.3 Spesifikasi *Use Case* Menampilkan Daftar Pantai

No.	Spesifikasi *Use case*	Deskripsi
1.	Melihat detail informasi pantai	Fitur untuk melihat detail informasi terkait pantai yang kita pilih.

Tabel 4.4 Spesifikasi *Use Case* Mendapatkan Rekomendasi

No.	Spesifikasi *Use case*	Deskripsi
1.	Memasukkan Bobot Kriteria	Fitur untuk memasukkan bobot kriteria objek wisata pantai.
2.	Melihat Hasil Rekomendasi	Menampilkan hasil rekomendasi dari system.

Tabel 4.5 Spesifikasi *Use Case* Menampilkan Informasi Pantai

No.	Spesifikasi *Use case*	Deskripsi
1.	Melihat detail informasi pantai	Fitur untuk melihat detail informasi terkait pantai yang kita pilih.

Tabel 4.6 Spesifikasi *Use Case* Menampilkan Peta

No.	Spesifikasi *Use case*	Deskripsi
1.	Melihat rute menuju tempat wisata	Fitur untuk menampilkan rute menuju tempat wisata pantai dengan menggunakan google maps.

4.1.4 Skenario Use Case

Skenario *usecase* berisi penjelasan yang lebih mendetail terkait spesifikasi *usecase*. Skenario *usecase* dapat dilihat pada Tabel 4.7 sampai Tabel 4.8.

Tabel 4.7 Skenario *Use Case* Menampilkan Daftar Pantai

Nama	**Menampilkan Daftar Pantai**	
Aktor	Pengguna Aplikasi	
Tujuan	Menampilkan daftar pantai yang berada pada database kedalam aplikasi.	
Deskripsi	*Use case* ini menjelaskan bagaimana pengguna melihat daftar pantai yang ada dalam database.	
Pre-condition	Aktor membuka aplikasi dan melewati tampilan *splash screen.*	
Post-condition	Menampilkan daftar pantai.	
Aliran Utama		
Aksi dari Aktor		**Respon dari Sistem**
1. Membuka aplikasi.		2. Menampilkan halaman *splash screen.*
		3. (a) Ada koneksi internet. Menampilkan halaman *home*.

	4. Aplikasi meminta data daftar pantai ke *web service* dengan alamat URL. Kemudian data ditampilkan pada halaman *home*
Aliran Alternatif	
Aksi dari Aktor	**Respon dari Sistem**
	3. (b) Tidak ada koneksi. Muncul peringatan bahwa tidak memiliki koneksi internet. Menu minta rekomendasi tidak akan aktif.

Tabel 4.8 Skenario *Use Case* Memasukkan Bobot Kriteria

Nama	**Memasukkan Kriteria**
Aktor	Pengguna Aplikasi
Tujuan	Untuk memasukkan bobot prioritas kriteria.
Deskripsi	*Use case* ini menjelaskan terkait bagaimana pengguna memasukkan bobot kriteria objek wisata pantai.
Pre-condition	Aktor harus membuka aplikasi terlebih dahulu. Setelah masuk ke halaman *home*, memilih menu minta rekomendasi. Kemudian muncul halaman *input* bobot kriteria untuk mengisi bobot kriteria.
Post-condition	Setelah meng-*input* semua data bobot kriteria, data tersebut akan disimpan sementara pada variabel sebelum dikirimkan ke *web service*.
Aliran Utama	
Aksi dari Aktor	**Respon dari Sistem**
1. Membuka aplikasi, masuk ke halaman *home* dan memilih menu minta rekomendasi.	2. Menampilkan halaman *input* bobot kriteria. Pada halaman terdapat 5 form isiin untuk bobot kriteria.
3. Mengisi form *input* bobot kriteria, kemudian tekan menu rekomendasi.	4. (a) Mengisi semua form. Semua *input* disimpan pada variable sementara dan proses *input* selesai.
Aliran Alternatif	
Aksi dari Aktor	**Respon dari Sistem**
	3. (b) Tidak mengisi semua form. Muncul peringatan (*toast*) bahwa ada form yang belum diisi/disimpan.

Tabel 4.9 Skenario *Use Case* Mendapatkan Rekomendasi

Nama	**Memasukkan Alternatif dan Kriteria**

Aktor	Pengguna Aplikasi
Tujuan	Untuk memberikan pembobotan pada kriteria sesuai dengan keinginan *user.*
Deskripsi	*Use case* ini menjelaskan bagaimana pengguna memasukkan bobot kriteria untuk mendapatkan rekomendasi tempat wisata pantai.
Pre-condition	Aktor sudah mengisi semua form pada halaman *input* bobot kriteria dan menekan minta rekomendasi
Post-condition	Menampilkan daftar rekomendasi pantai.
Aliran Utama	
Aksi dari Aktor	**Respon dari Sistem**
1. Menekan minta rekomendasi setelah semua form pada halaman *input* bobot kriteria terisi.	2. (a) Ada koneksi internet dan sensor GPS menyala. Data *input* dikirim ke *web service* melalui URL untuk dihitung metode TOPSIS*.* Kemudian hasil rekomendasi berformat JSON dikirim kembali ke aplikasi perangkat bergerak dari *web service*.
	3. Menampilkan halaman hasil rekomendasi yang berisi daftar hasil rekomendasi yang direkomendasikan sistem.

Tabel 4.10 Skenario *Use Case* Menampilkan Informasi Pantai

Nama	**Melihat Ringkasan**
Aktor	Pengguna Aplikasi
Tujuan	Menampilkan informasi terkait pantai yang telah dipilih oleh pengguna.
Deskripsi	*M*enjelaskan terkait bagaimana pengguna melihat informasi terkait pantai yang telah dipilih pengguna dari hasil rekomendasi system.
Pre-condition	Pada halaman yang menampilkan daftar hasil rekomendasi pantai, pengguna memilih salah satu pantai dari hasil rekomendasi.
Post-condition	Menampilkan halaman informasi pantai.
Aliran Utama	
Aksi dari Aktor	**Respon dari Sistem**
1. Menekan salah satu pantai hasil rekomendasi system.	2. Menampilkan halaman *summary* yang berisi data alternatif, kriteria, dan koordinat pengguna.

Tabel 4.11 Skenario *Use Case* Menampilkan Peta

Nama *Use Case*	**Melihat Hasil Rekomendasi**
Aktor	Pengguna Aplikasi
Tujuan	Untuk menampilkan peta digital dan juga rute menuju tempat wisata pantai.
Deskripsi	*Use case* ini menjelaskan bagaimana pengguna melihat peta digital dan rute terkait pantai yang telah dipilih pengguna dari hasil rekomendasi system.
Pre-condition	Pengguna telah memilih salah satu dari pantai hasil rekomendasi system.
Post-condition	Menampilkan peta Google Maps yang berisi rute menuju lokasi pantai yang dipilih.
Aliran Utama	
Aksi dari Aktor	**Respon dari Sistem**
1. Menekan *button* yang berfungsi untuk menampilkan peta	2. Menampilkan Google Maps sekaligus rute menuju lokasi pantai.

4.1.4 Analisis Kebutuhan Non Fungsional

Merupakan kebutuhan yang secara langsung tidak diminta oleh pengguna. Detail terkait kebutuhan non fungsional terdapat pada Tabel 4.8.

Tabel 4.12 Spesifikasi Kebutuhan Non Fungsional

Identifier	**Parameter**	**Deskripsi**
SKNF-001	Validasi Algoritma	Memvalidasi hasil keluaran sistem dengan hasil perhitungan manual. Jika hasil perhitungan sama, maka penerapan metode TOPSIS dikatakan berhasil.
SKNF-002	*Rank Consistensy*	Mengetahui seberapa konsisten hasil dari rekomendasi pantai.

4.2 Perancangan Perangkat Lunak

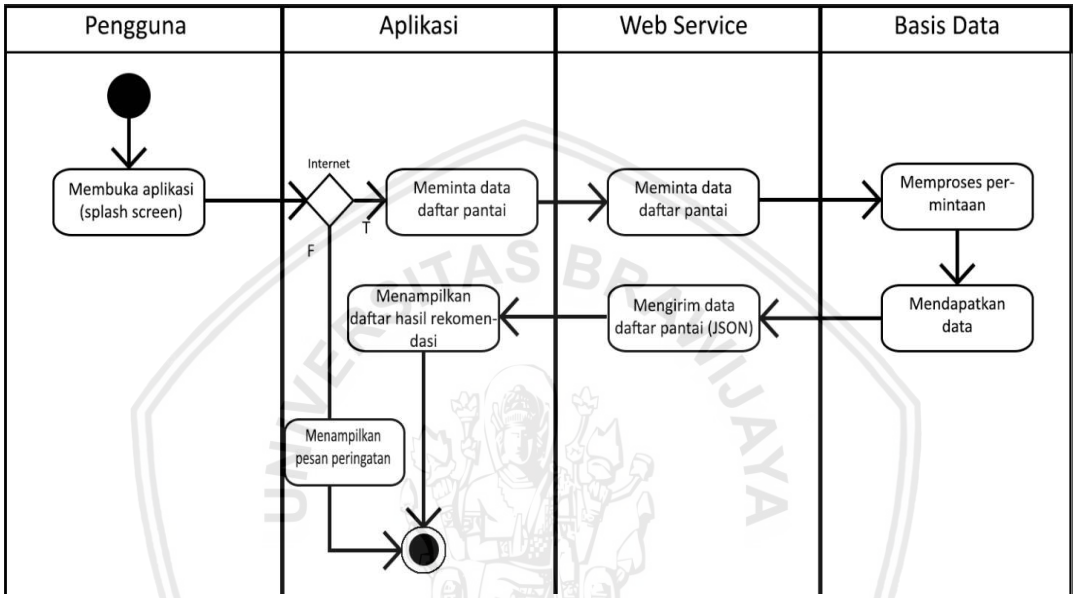
Penjelasan terkait perancangan yang akan dibangun merupakan isi dari tahapan ini. Perancangan pada penelitian ini terdiri dari beberapa tahap, yaitu: arsitektur sistem, *activity diagram, class diagram, sequence diagram*, algoritma, basis data, lalu perancangan navigasi dan antarmuka.

Terdapat 2 tahapan pada rancangan arsitektur sistem, yaitu: perancangan arsitektur aplikasi dan akses data eksternal. Selanjutnya terdapat 2 tahap dalam perancangan algoritma, yakni: algoritma TOPSIS, dan perhitungan manual. Terakhir terdapat 2 tahap pada perancangan navigasi dan antarmuka, yakni: perancangan *screen flow* dan antarmuka.

4.2.1 Perancangan *Activity Diagram*

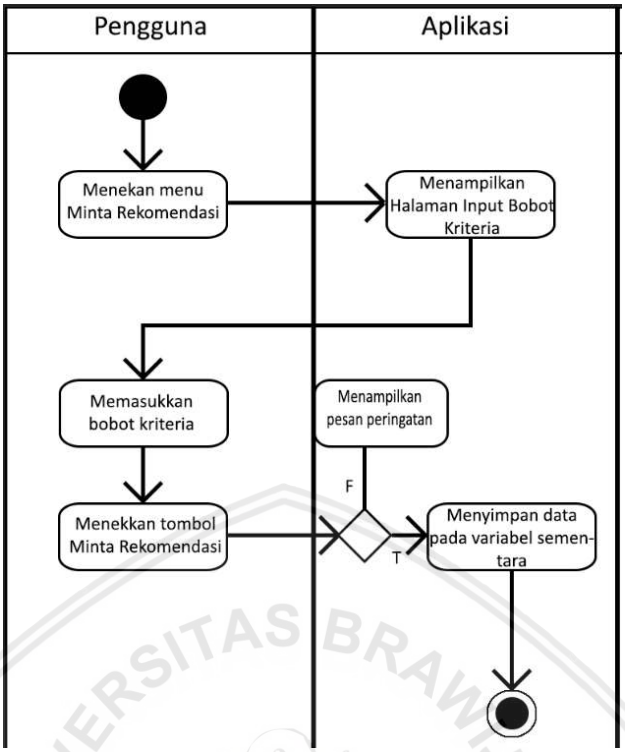
Melakukan pemodelan berdasarkan pada *scenario usecae* yang sudah dibuat sebelumnya merupakan isi pada tahapan ini. Gambar 4.4 sampai Gambar 4.8 merupakan representasi *activity diagram*.

Gambar 4.4 pengguna membuka aplikasi, kemudian system akan menampilkan halaman *splash screen* dan halaman *home* dimana secara bersamaan aplikasi meminta data pantai yang berada pada *database* melalui *web service*.



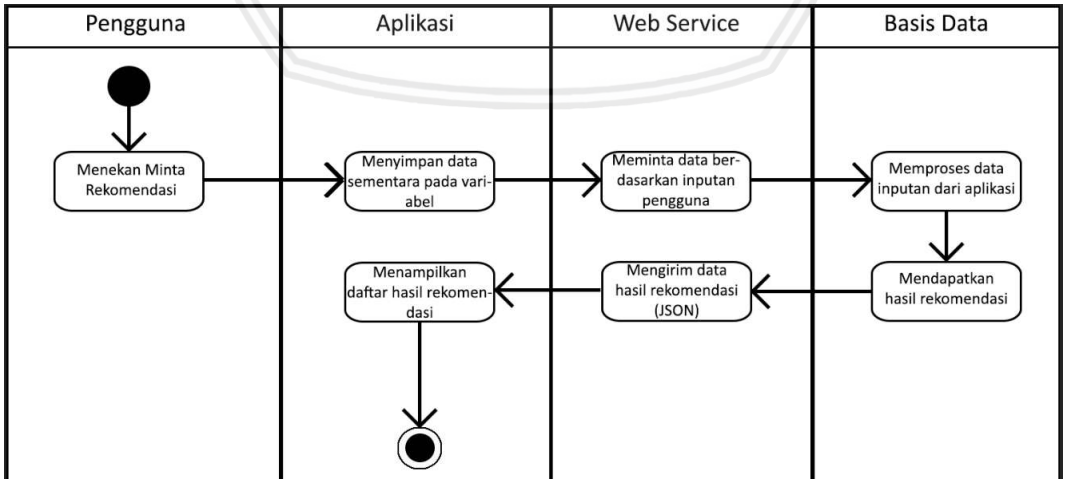
Gambar 4.4 *Activity Diagram* Menampilkan Daftar Pantai

Pada Gambar 4.5 pengguna harus sudah masuk kedalam halaman *home* dan pengguna memilih menu minta rekomendasi. Sistem akan masuk ke halaman untuk *input* bobot kriteria, setelah nilai dari bobot diisi dan menekan tombol OK, maka sistem akan menyimpan data masukan pada variabel sementara.



Gambar 4.5 *Activity Diagram* Memasukkan Bobot Kriteria

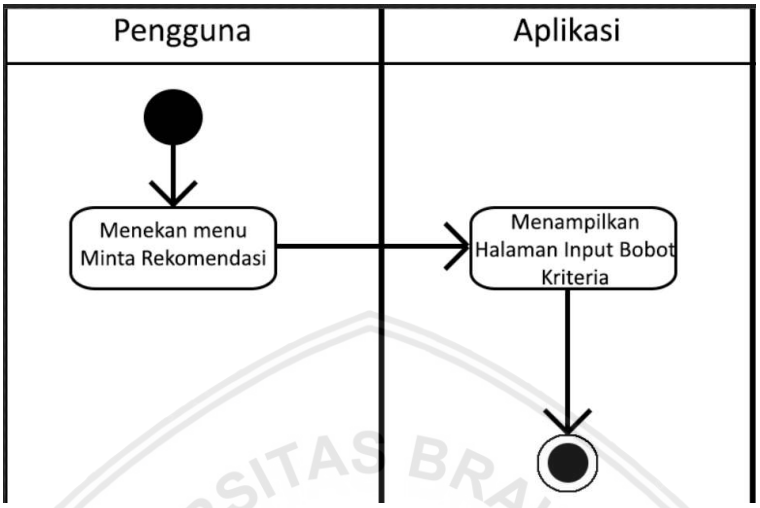
Pada Gambar 4.6 pengguna harus sudah masuk pada halaman *input* bobot kriteria, mengisi semua form dan menekan minta rekomendasi. Aplikasi akan mengirim data masukkan pengguna ke *web service* sekaligus meminta data hasil rekomendasi. Berdasarkan masukan dari aplikasi perangkat bergerak pengguna, hasil rekomendasi dari algoritma TOPSIS akan dikirim kembali ke aplikasi perangkat bergerak pengguna melalui *web service* juga. Pada aplikasi perangkat bergerak pengguna, akan ditampilkan daftar hasil rekomendasi.



Gambar 4.6 *Activity Diagram* Melihat Hasil Rekomendasi

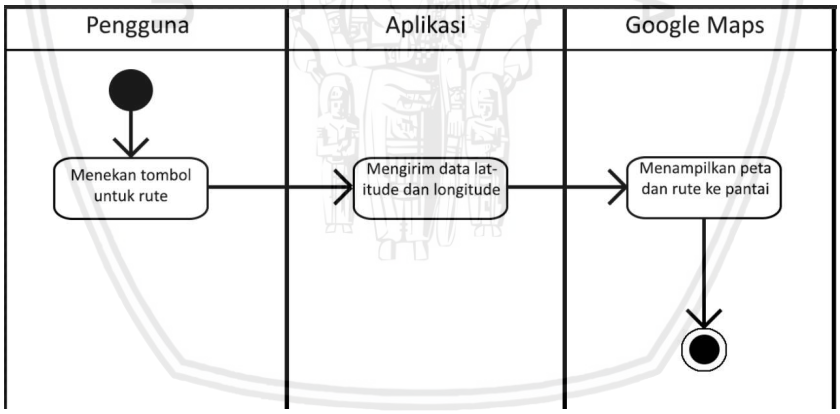
Pada Gambar 4.7 pengguna harus sudah mendapatkan hasil rekomendasi pantai oleh system dengan menggunakan menu minta rekomendasi pantai

terlebih dahulu. Dari daftar hasil rekomendasi tersebut pengguna bisa memilih salah satu dari daftar yang ada. Ketika pengguna memilih salah satu daftar pantai, maka system akan menampilkan halaman informasi pantai pilihan pengguna aplikasi perangkat bergerak.



Gambar 4.7 *Activity Diagram* Menampilkan Informasi Pantai

Pada Gambar 4.6 pengguna harus menekan tombol pada halaman detail pantai, kemudian sistem akan membuka aplikasi *google maps,* lalu aplikasi akan menunjukan rute menuju lokasi pantai.



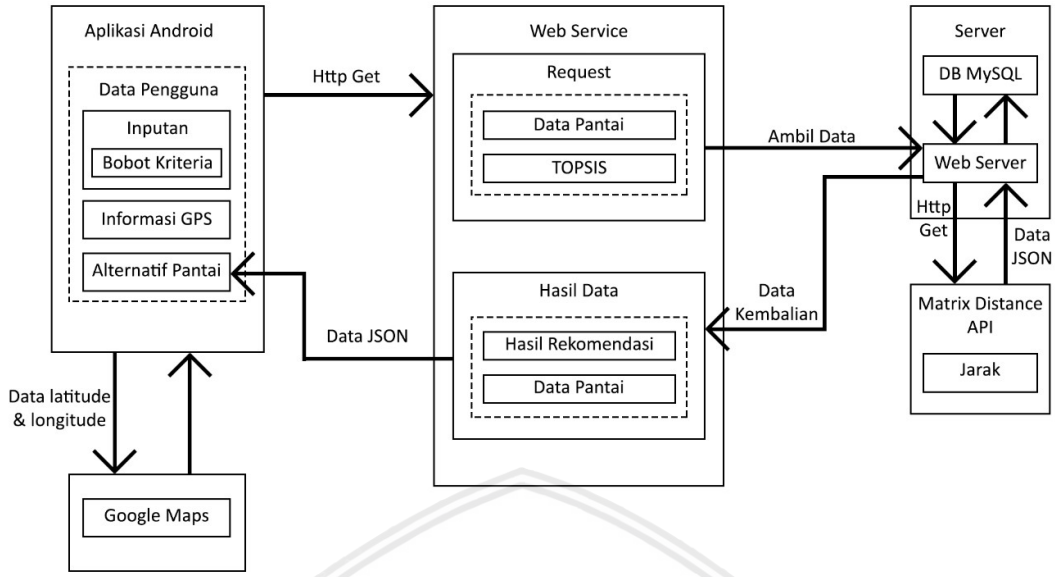
Gambar 4.8 *Activity Diagram* Menampilkan Peta

4.2.2 Perancangan Arsitektur Sistem

Penjelasan terkait arsitektur yang akan dibangun akan dijelaskan pada bagian ini. Bagian ini terbagi menjadi 2, yaitu perancangan arsitektur perangkat bergerak dan akses data eksternal.

4.2.4 Perancangan Arsitektur Perangkat Bergerak

Terdapat 3 bagian arstitektur perangkat bergerak, yakni: bagian aplikasi perangkat bergerak, *web service*, dan *server.* Pada Gambar 4.9 terdapat gambar arsitektur perangkat bergerak secara utuh.

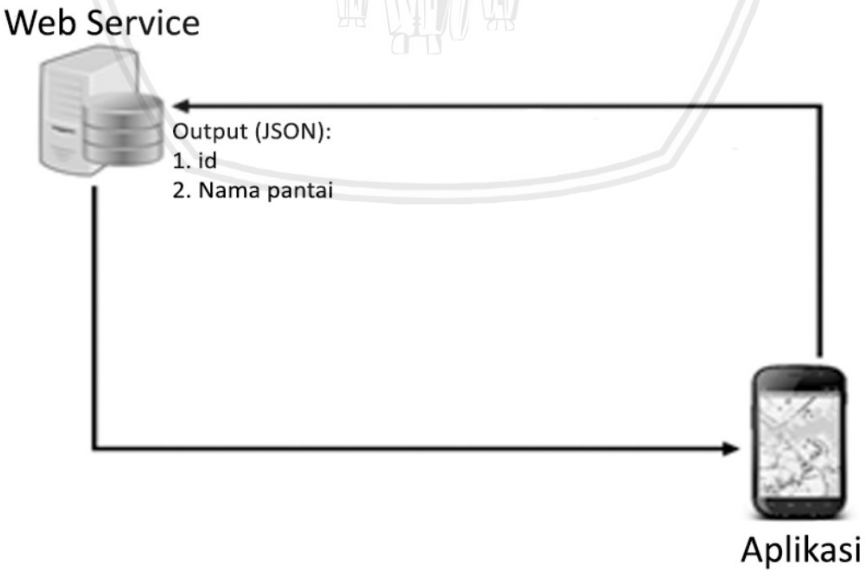


Gambar 4.9 Arsitektur Perangkat Bergerak

4.2.4 Perancangan Akses Data Eksternal

Perancangan ini menjelaskan bagaimana menentukan *input* yang diperlukan untuk menghasilkan *output* berupa data tertentu yang dibutuhkan oleh aplikasi perangkat bergerak. *Input* dan *output* yang dimaksud pada perancangan akses data eksternal ditunjukkan dalam Gambar 4.10 sampai Gambar 4.11.

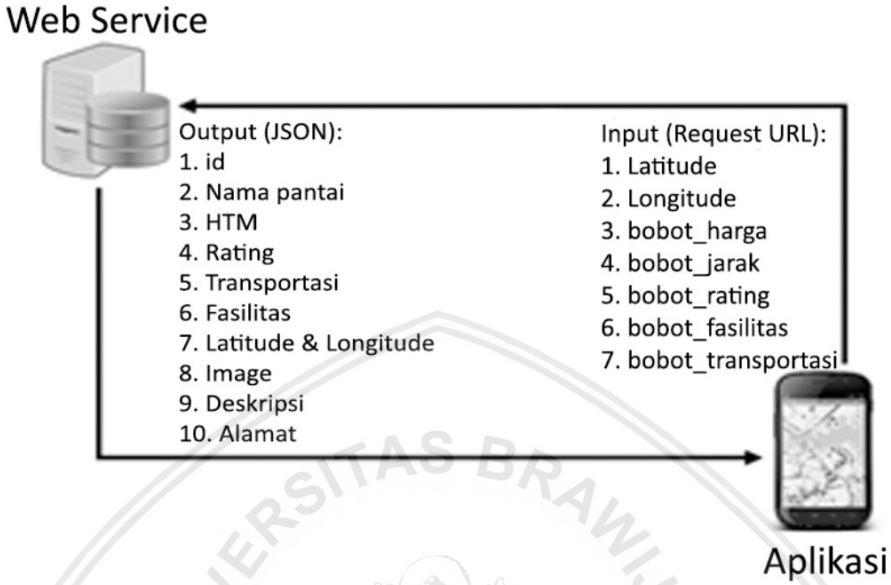
Gambar 4.10 menjelaskan akses data JSON daftar pantai yang berada pada database tanpa menginputkan data dari aplikasi kepada *web service* dapat menghasilkan *output* daftar nama pantai yang digunakan dan ditampilkan di aplikasi. Hasil dari *output* berformat JSON berupa nama pantai.



Gambar 4.10 Akses Data Eksternal JSON Daftar Pantai

Gambar 4.11 menjelaskan akses data JSON hasil rekomendasi pantai dengan *input* nilai bobot kriteria harga tiket masuk, jarak, fasilitas, transportasi, rating,

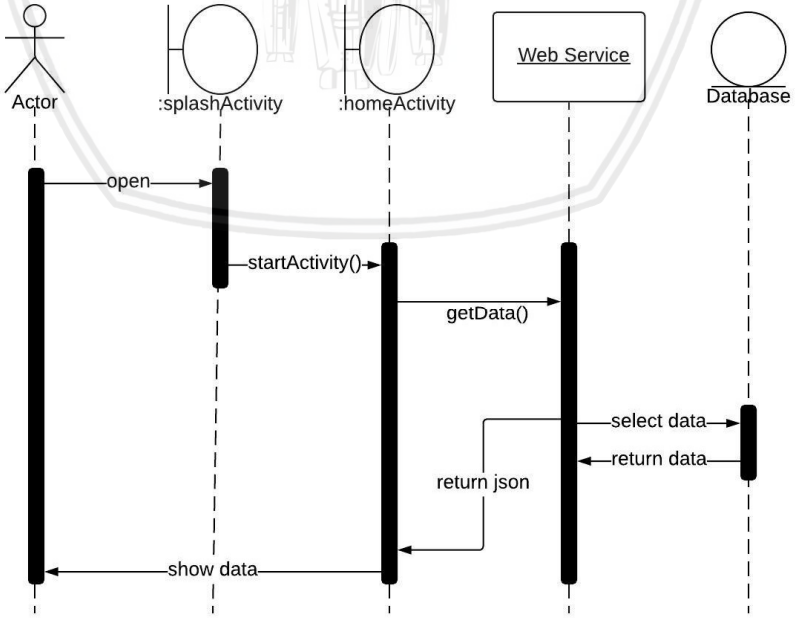
latitude dan *longitude* dapat menghasilkan beberapa *output* yang digunakan dan ditampilkan di aplikasi. Hasil dari *output* berformat JSON berupa kode pantai, nama pantai, fasilitas, transportasi, rating, deskripsi pantai, gambar pantai, dan alamat.



Gambar 4.11 Akses Data Eksternal JSON Hasil Rekomendasi Pantai

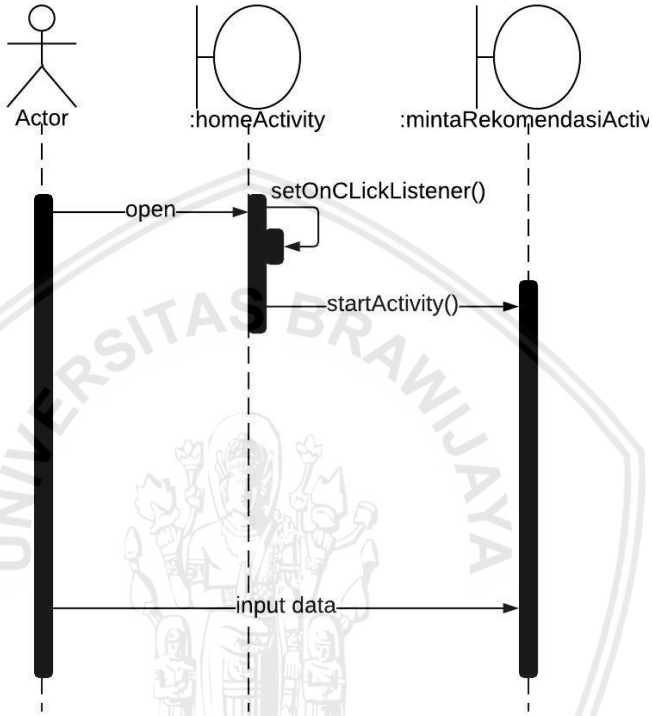
4.2.3 Perancangan *Sequence Diagram*

Penjelasan terkait interaksi dan hubungan antara objek pada system yang akan dibangun merupakan isi dari perancangan *sequence diagram*.



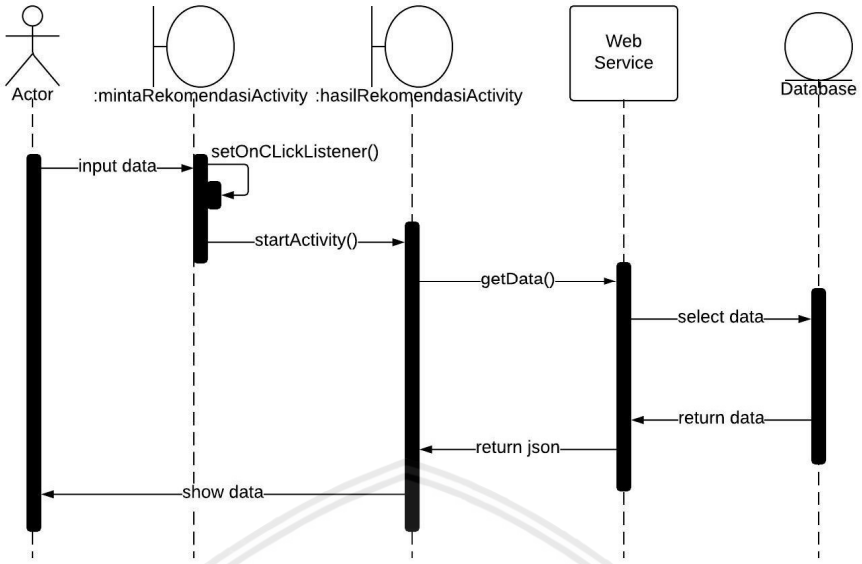
Gambar 4.12 *Sequence Diagram* Menampilkan Daftar Pantai

Sequence diagram yang menjelaskan interaksi untuk menampilkan daftar pantai terdapat pada Gambar 4.12. Pertama memanggil kelas *SplashActivity* kemudian memanggil kelas *HomeActivity.* Selanjutnya akan dijalankan method getData() untuk meminta data daftar pantai pada database melalui *web service.*



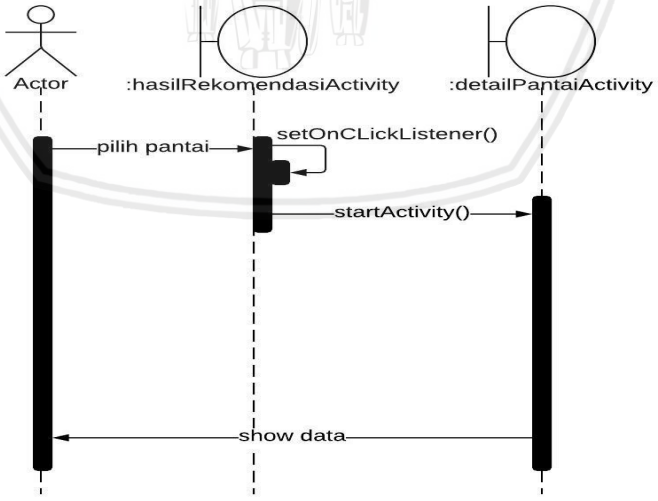
Gambar 4.13 *Sequence Diagram* Memasukan Bobot Kriteria

Gambar 4.13 menampilkan *sequence diagram* memasukan bobot kriteria skenario *usecase* memasukkan bobot kriteria pada Tabel 4.8. Setelah masuk kedalam *homeActivity,* pengguna menekan menu minta rekomendasi untuk memanggil kelas mintaRekomendasiActivity. Pada kelas ini terdapat 5 form yang harus diisi oleh pengguna.



Gambar 4.14 *Sequence Diagram* Melihat Hasil Rekomendasi

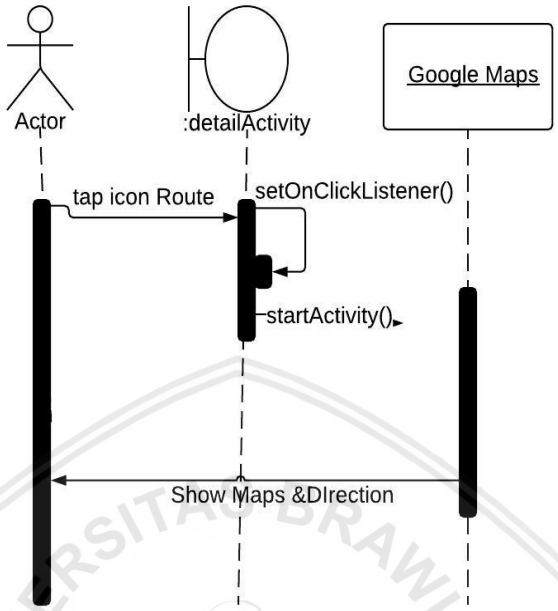
Gambar 4.14 menampilkan *sequence diagram* memasukkan alternatif dan kriteria sesuai dengan skenario *usecase* melihat hasil rekomendasi pada Tabel 4.9. Setelah kita mengisi semua form pada kelas mintaRekomendasiActivity, pengguna menekan tombol minta rekomendasi. Tombol tersebut berfungsi untuk memanggil kelas hasilRekomendasiActivity, dimana kelas tersebut akan menjalankan *method getData()* yang berfungsi untuk mengirim data sekaligus meminta data hasil rekomendasi.



Gambar 4.15 *Sequence Diagram* Menampilkan Informasi Pantai

Sequence diagram yang menjelaskan interaksi untuk menampilkan informasi pantai terdapat pada Gambar 4.15. Setelah melihat hasil rekomendasi, pengguna

dapat memilih salah satu pantai yang ada. Dimana akan memanggil kelas detailPantaiActivity yang menampilkan informasi terkait pantai yang dipilih.

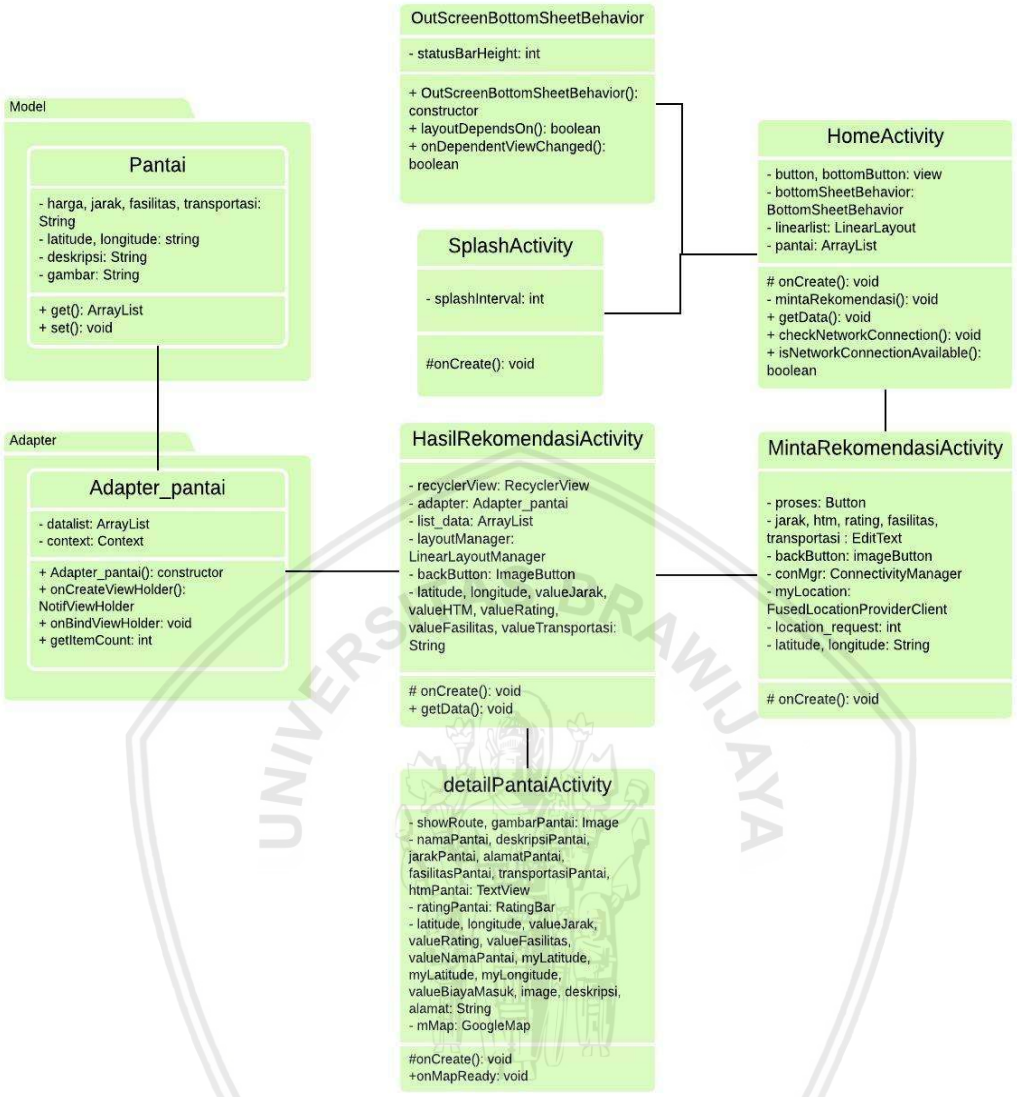


Gambar 4.16 *Sequence Diagram* Menampilkan Peta

Sequence diagram yang menjelaskan interaksi untuk menampilkan peta terdapat pada Gambar 4.16. Pada detailPantaiActivity terdapat tombol yang berfungsi untuk membuka aplikasi *google maps*.

4.2.4 Perancangan *Class Diagram*

Dapat menampilkan kelas-kelas yang akan ada pada system yang akan dibangun merupakan kegunaan dari perancangan ini. Pada perancangan *class diagram* akan dipresentasikan hubungan antar kelas dan cara saling terhubungnya.



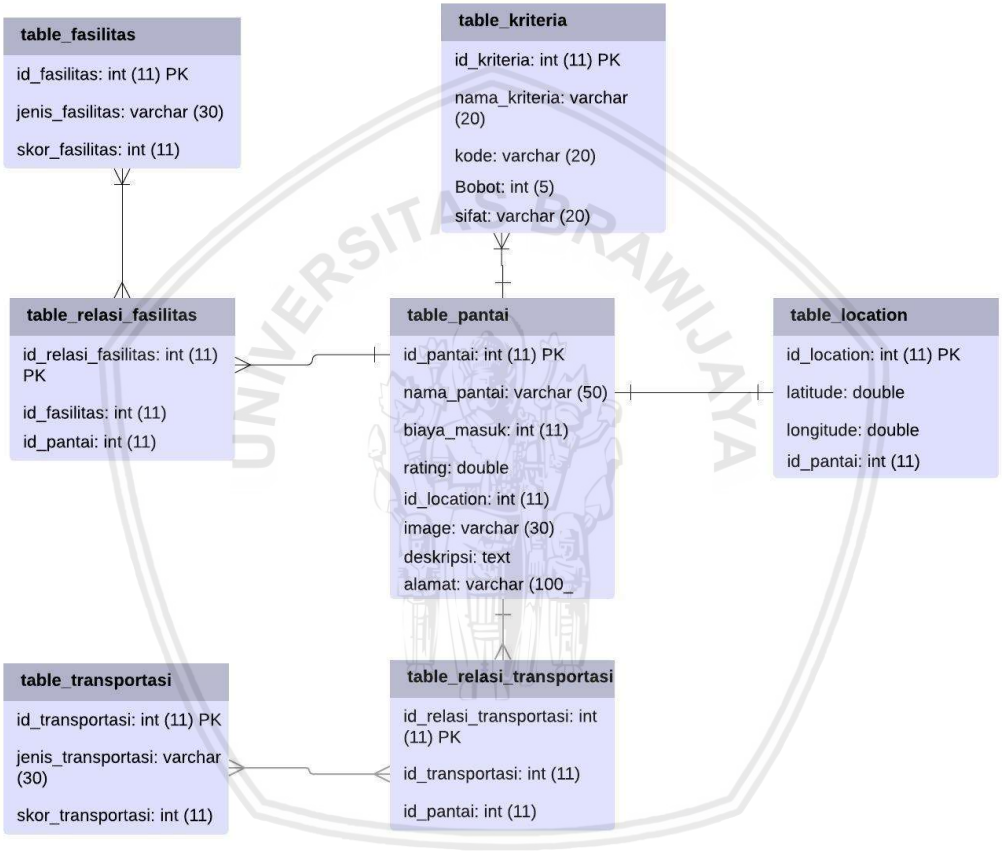
Gambar 4.17 *Class Diagram*

Diagram kelas secara utuh dari system yang akan dibuat dapat dilihat pada Gambar 4.17. Kelas SplashActivity merupakan kelas yang akan pertama kali dijalankan ketika aplikasi dibuka dimana berfungsi untuk menampilkan splashscreen. Setelah itu ada kelas HomeActivity yang merupakan kelas yang terhubung dengan halaman utama. Pada HomeActivity terdapat *button* yang menghubunkan dengan kelas mintaRekomendasiActivity. Pada kelas mintaRekomendasiActivity terdapat form yang berfungsi untuk masukan data bobot kriteria. Selain itu, terdapat tombol minta rekomendasi yang menghubungkan dngan kelas hasilRekomendasiActicity. Kelas hasilRekomendasiActivity merupakan kelas yang akan menjadi pemicu sekaligus penyambung aplikasi kepada *web service* agar aplikasi dapat mendapatkan rekomendasi pantai. Selanjutnya terdapat kelas detailPantaiActivity dimana kelas ini merupakan kelas yang berfungsi untuk menampilkan informasi terkait pantai-

pantai yang ada. Serta untuk bisa menampilkan rute menuju lokasi pantai maka kelas detailPantaiActivity dapat memicu agar aplikasi *google maps* terbuka.

4.2.5 Perancangan Basis Data

Dapat mempresentasikan bagaimana cara penyimpanan data pada system merupakan kegunaan dari perancangan ini. Perancangan ini digambarkan dalam bentuk diagram hubungan antar entitas yang di dalamnya.



Gambar 4.18 Perancangan Basis Data

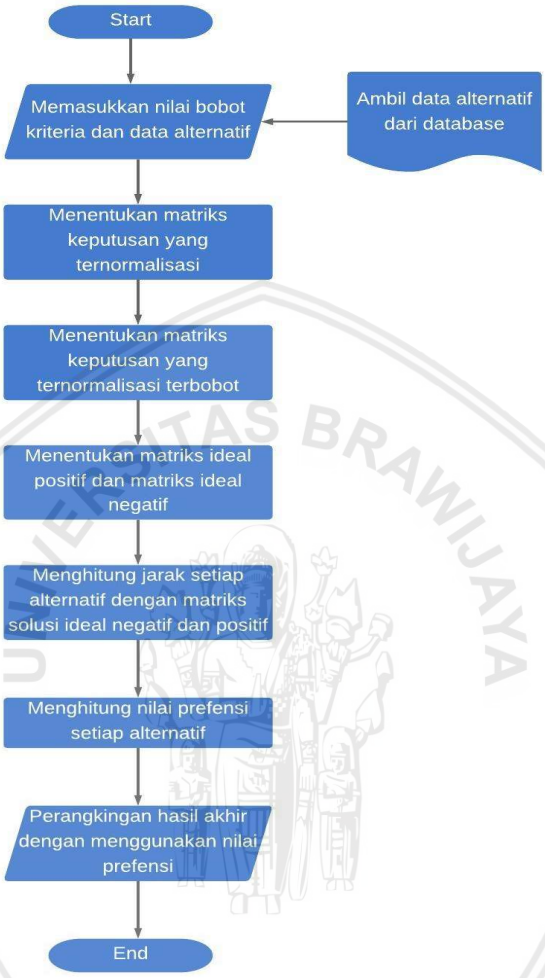
Penjelasan terkait perancangan basis data yang akan diterapkan pada system dapat dilihat pada Gambar 4.18. Terdapat tujuh entitas/tabel yang digunakan, yaitu: table_pantai, table_fasilitas, table_relasi_fasilitas, table_kriteria, table_location, table_transportasi dan table_relasi_transportasi.

4.2.6 Perancangan Algoritma

Perancangan ini menjelaskan algoritma yang digunakan untuk merekomendasikan tempat wisata pantai. Perancangan ini berupa perancangan algoritma TOPSIS, dan perhitungan manual.

4.2.6.1 Perancangan TOPSIS

Dapat memberikan hasil rekomendasi dari alternate yang ada berdasarkan bobot kriteria yang dimasukkan merupakan fungsi dari metode TOPSIS.



Gambar 4.19 *Flowchart* TOPSIS

Perhitungan dan alur yang harus dilalui pada metode TOPSIS dapat dilihat pada Gambar 4.19. Detail dan contoh perhitungan dari tiap bagian *flowchart* akan dijelaskan pada sub bab 4.2.6.2 Perhitungan Manual.

4.2.6.2 Perhitungan Manual

Perhitungan algoritma TOPSIS dapat dijelaskan pada perhitungan manual sebagai berikut:

- **Perhitungan TOPSIS**

a. Menentukan kriteria dan bobot

Tabel 4.10 menjelaskan terkait kriteria dan bobot kriteria apa saja yang akan digunakan untuk menentukkan nilai bobot prioritas untuk merangking alternatif tempat wisata pantai sebagai hasil rekomendasi nantinya. Kode P mengartikan

harga/*price,* kode D mengartikan jarak/*distance,* kode R mengartikan *rating,* kode T mengartikan transportasi dan kode F mengartikan fasilitas.

Tabel 4.13 Sampel Bobot Kriteria

No	Kriteria	Status	Bobot
1	P	COST	1
2	D	COST	2
3	R	Benefit	2
4	T	Benefit	1
5	F	Benefit	1

b. Menentukan data alternatif

Contoh data alternatif yang digunakan untuk perankingan menggunakan metode TOPSIS dapat dilihat pada Tabel 4.16.

Tabel 4.14 Sampel Data Alternatif

No	Alternatif Pantai	P	D	R	T	F
1	Pantai Balekambangan	10000	61	4,3	3	6
2	Pantai Ngantep	4000	59	4,3	2	4
3	Pantai Bajulmati	3000	58	4,3	3	3
4	Pantai Goa Cina	4000	72	4,3	3	4
5	Pantai Sendang Biru	6000	74	4,1	3	4

c. Menghitung normalisasi matriks alternatif

Nilai normalisasi matriks alternatif dapat dihitung dari nilai matriks pada Tabel 4.14 dan menggunakan Persamaan (2.7). Pertama, melakukan penjumlahan nilai alternatif tiap kriteria yang sudah dipangkat dua, selanjutnya lakukan proses akar kuadrat. Langkah selanjutnya adalah dengan melakukan pembagian pada tiap *cell* dengan hasil jumlah alternatif tiap kriteria. Hasil penjumlahan nilai tiap alternatif per kriteria dan perhitungan normalisasi matriks alternatif dapat dilihat pada Tabel 4.15 dan Tabel 4.16 dimana mengacu pada Persamaan 2.1.

Tabel 4.15 Hasil Penjumlahan Alternatif Tiap Kriteria

No	Alternatif Pantai	P	D	R	T	F
1	Pantai Balekambangan	10000	61	4,3	3	6
2	Pantai Ngantep	4000	59	4,3	2	4
3	Pantai Bajulmati	3000	58	4,3	3	3
4	Pantai Goa Cina	4000	72	4,3	3	4

5	Pantai Sendang Biru	6000	74	4,1	3	4
		13304,13	145,7	9,5	6,3	9,6

Berikut contoh perhitungan untuk menghitung pembagi nilai alternatif tiap kriteria Kolom P: $\sqrt{10000^2 + 4000^2 + 3000^2 + 4000^2 + 6000^2} = 13304{,}13$

Tabel 4.16 Hasil Normalisasi Matriks Alternatif

No	Alternatif Pantai	P	D	R	T	F
1	Pantai Balekambangan	0,752	0,42	0,45	0,47	0,62
2	Pantai Ngantep	0,3	0,4	0,45	0,32	0,41
3	Pantai Bajulmati	0,22	0,39	0,45	0,47	0,31
4	Pantai Goa Cina	0,3	0,49	0,45	0,47	0,41
5	Pantai Sendang Biru	0,45	0,51	0,43	0,47	0,41

Berikut contoh perhitungan untuk menghitung normalisasi matrik alternatif.

- Baris 1, Kolom P: $\frac{10000}{13304{,}13} = 0{,}752$

d. Menghitung normaliasi matriks terbobot

Normalisasi matriks terbobot dihitung sesuai dengan Persamaan (2.8), dimulai dengan mengalikan nilai matriks pada Tabel 4.16 dengan nilai pada Tabel 4.14. Hasil perhitungan dapat dilihat pada Tabel 4.17 dimana mengacu pada Persamaan 2.2.

Tabel 4.17 Hasil Normaliasi Matriks Terbobot

No	Alternatif Pantai	P	D	R	T	F
1	Pantai Balekambangan	0,752	0,84	0,90	0,47	0,62
2	Pantai Ngantep	0,3	0,8	0,90	0,32	0,41
3	Pantai Bajulmati	0,22	0,75	0,90	0,47	0,31
4	Pantai Goa Cina	0,3	0,98	0,90	0,47	0,41
5	Pantai Sendang Biru	0,45	1,02	0,86	0,47	0,41

Berikut contoh perhitungan untuk menghitung normalisasi matrik terbobot antara nilai alternatif tiap kriteria dengan nilai bobot prioritas tiap kriteria.

- Baris 1, Kolom P: $0{,}752 * 1 = 0{,}752$

e. Menghitung solusi ideal

Proses perhitungan sesuai dengan Persamaan (2.9) dan (2.10) dari matriks normalisasi terbobot pada Tabel 4.17. Untuk kriteria menguntungkan, solusi ideal positif adalah nilai tertinggi dari seluruh alternatif pada kriteria yang sama dan solusi ideal negatif merupakan nilai terendah dari seluruh alternatif pada kriteria yang sama. Sedangkan kriteria dengan sifat merugikan, solusi ideal positif merupakan nilai terendah dari seluruh alternatif pada kriteria yang sama dan solusi ideal negatif merupakan nilai tertinggi dari seluruh alternatif pada kriteria yang sama.

Tabel 4.18 Hasil Solusi Ideal Positif dan Solusi Ideal Negatif

No	Alternatif Pantai	P	D	R	T	F
1	A+	0,22	0,75	0,90	0,47	0,62
2	A-	0,752	1,02	0,86	0,32	0,31

Hasil penentuan solusi ideal dapat dilihat pada Tabel 4.20 dimana untuk menghitung A+ mengacu pada persamaan 2.3 dan A- mengacu pada persamaan 2.4. Contoh perhitungan untuk menemukan solusi ideal adalah sebagai berikut.

- Baris 1, Kolom P:

$$\min\{0,752;0,3;0,22;0,3;0,45\} = 0,22$$

- Baris 2, Kolom P:

$$\max\{0,752;0,3;0,22;0,3;0,45\} = 0,752$$

f. Menghitung jarak alternatif terhadap solusi ideal

Jarak alternatif terhadap solusi ideal dihitung sesuai dengan Persamaan (2.11) dan (2.12) dan menggunakan nilai normalisasi matriks terbobot pada Tabel 4.17 dan solusi ideal pada Tabel 4.18. Hasil perhitungan jarak alternatif terhadap solusi ideal positif dan solusi ideal negatif dapat dilihat pada Tabel 4.19 dimana mengacu pada Persamaan 2.5 dan Persamaan 2.6.

Tabel 4.19 Hasil Perhitungan Jarak Terhadap Solusi Ideal Positif dan Negatif

No	Jarak	
	D+	D-
1	0.539559079	0.3906405
2	0.274772633	0.514105048
3	0.31	0.616460867
4	0.321558704	0.489902031
5	0.414125585	0.351715794

Berikut contoh perhitungan untuk mencari jarak alternatif terhadap solusi ideal.

- Baris 1, Kolom D+:

$$\sqrt{(0,22-0,752)^2+(0,75-0,84)^2+(0,9-0,9)^2+(0,47-0,47)^2+(0,62-0,62)^2} = 0.54$$

- Baris 1, Kolom D-:

$$\sqrt{(0,752-0,752)^2+(0,84-1,02)^2+(0,9-0,86)^2+(0,47-0,32)^2+(0,62-0,31)^2} = 0.39$$

g. Menghitung nilai preferensi

Nilai preferensi dihitung sesuai dengan Persamaan (2.13) dari hasil jarak alternatif terhadap solusi ideal pada Tabel 4.19. Hasil perhitungan dapat dilihat pada Tabel 4.20 dimana mengacu pada Persamaan 2.7.

Tabel 4.20 Hasil Perhitungan Nilai Prefrensi

No	Alternatif Pantai	Preferensi
1	Pantai Balekambangan	0.419953
2	Pantai Ngantep	0.651692
3	Pantai Bajulmati	0.665393
4	Pantai Goa Cina	0.603729
5	Pantai Sendang Biru	0.459254

Berikut contoh perhitungan.

- Baris 1: $\frac{0.3906405}{(0.3906405+0.539559079)} = 0.419953$

h. Perangkingan nilai preferensi

Langkah selanjutnya adalah dengan melakukan pengurutan seperti yang ditunjukkan pada Tabel 4.21.

Tabel 4.21 Hasil Perangkingan

No	Alternatif Pantai	Preferensi
1	Pantai Bajulmati	0.665393
2	Pantai Ngantep	0.651692
3	Pantai Goa Cina	0.603729
4	Pantai Sendang Biru	0.459254
5	Pantai Balekambangan	0.419953

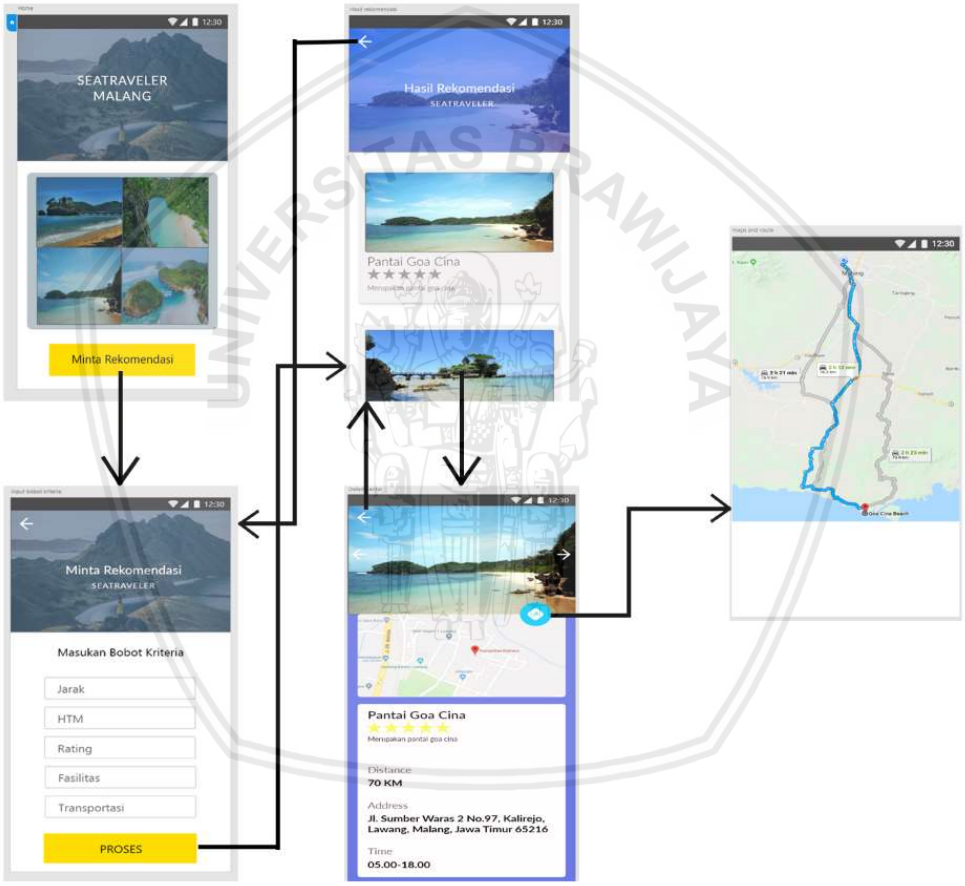
Hasil pengurutan nilai preferensi ditunjukkan pada Tabel 4.21. Maka pantai bajulmati dengan nilai 0.665393 merupakan pantai yang paling di rekomendasikan oleh sistem.

4.2.7 Perancangan Navigasi dan Antarmuka

Penjelasan terkait bagaimana rancangan antarmuka dari setiap halaman yang ada dan juga hubungan antara tiap halaman dijelaskan pada bagian ini. Perancangan ini juga memberikan gambaran terkait alur dan tahapan-tahapan halaman yang ada pada aplikasi.

4.2.7.1 Perancangan Screen Flow

Penjelasan terkait alur pada system merupakan isi pada perancangan *screen flow.*



Gambar 4.20 *Screen Flow* Aplikasi Perangkat Bergerak

Gambar 4.20 merupakan gambaran dari navigasi dalam *screen flow* dari masing-masing halaman. *Screen flow* ini memiliki 5 tampilan halaman yang akan ada pada aplikasi rekomendasi tempat wisata pantai. Kelima halaman tersebut terdiri dari: halaman *home* yang berada pada kiri atas, halaman input bobot kriteria yang berada pada kiri bawah, halaman hasil rekomendasi yang berada pada kanan atas, halaman detail pantai yang berada pada kanan bawah dan halaman *route map* yang berada pada ujung kanan.

4.2.7.2 Perancangan Antarmuka

Gambar 4.21 sampai Gambar 4.25 merupakan hasil perancangan dari system.

a. Halaman *Home*

Gambar 4.21 Antarmuka Halaman *Home*

Home merupakan halaman yang akan muncul pada setelah splashscreen ketika pengguna membuka aplikasi. Di halaman *home* terdapat *button* Minta Rekomendasi yang berfungsi untuk berpindah ke halaman untuk mengisi bobot kriteria.

b. Halaman *Input* Bobot Kriteria

Input Bobot Kriteria merupakan halaman yang digunakan untuk menginput bobot kriteria. Di halaman ini akan ada 5 form yang dapat diisi oleh pengguna, form tersebut antara lain: form bobot kriteria jarak, form bobot kriteria HTM (Harga Tiket Masuk), form bobot kriteria *rating,* form bobot kriteria fasilitas dan

form bobot kriteria transportasi. Gambar 4.19 menampilkan rancangan *input* bobot kriteria.

Gambar 4.22 Antarmuka Halaman *Input* Bobot Kriteria

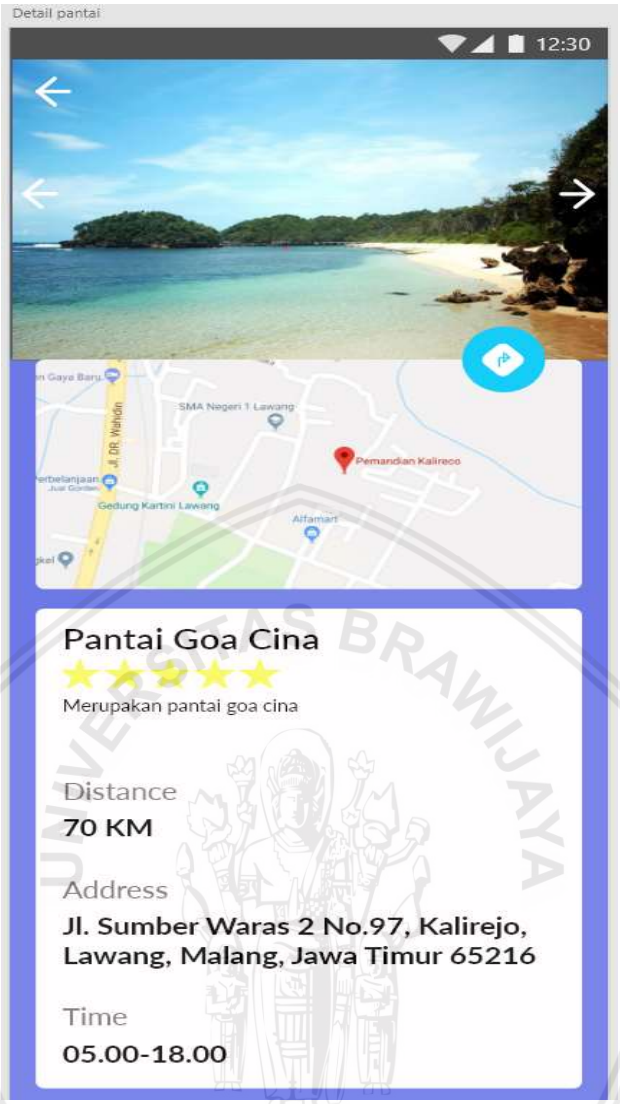
c. Halaman Hasil Rekomendasi

Gambar 4.23 Antarmuka Halaman Hasil Rekomendasi

Hasil rekomendasi halaman yang digunakan agar dapat melihat hasil rekomendasi tempat wisata pantai. Di halaman hasil rekomendasi menampilkan daftar hasil rekomendasi pantai yang direkomendasikan oleh sistem. Gambar 4.20 menampilkan rancangan antarmuka halaman hasil rekomendasi.

d. Halaman Detail Pantai

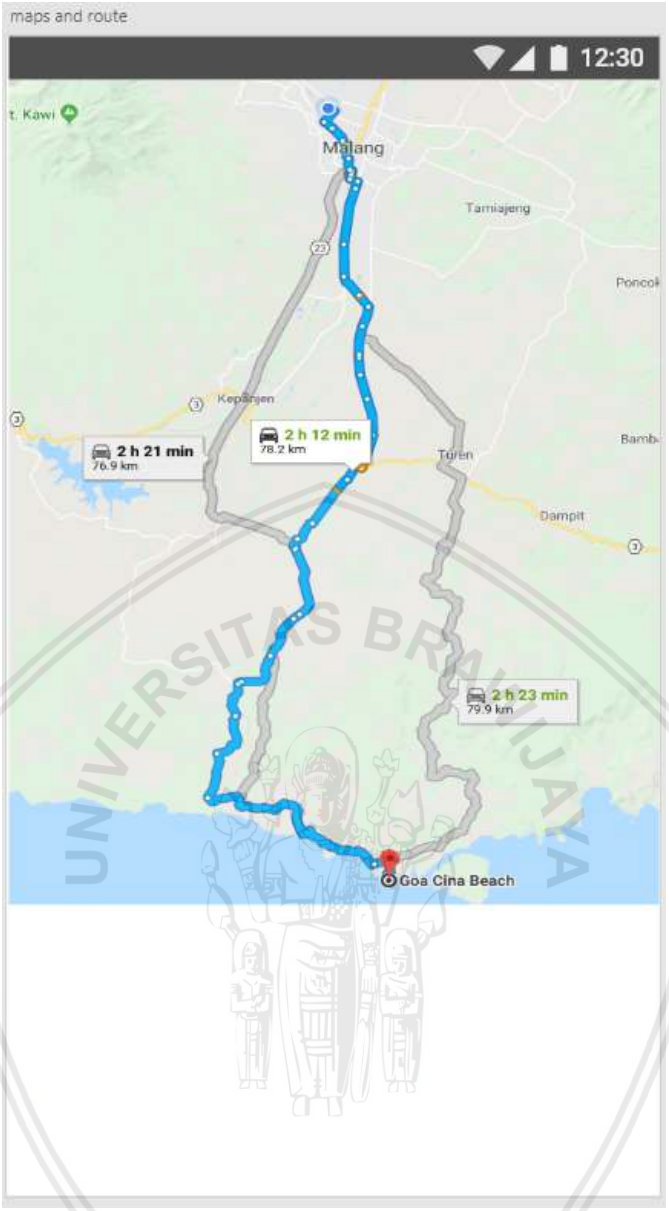
Detail pantai merupakan halaman yang digunakan untuk melihat informasi terkait pantai yang dipilih oleh pengguna secara lebih mendetail. Gambar 4.21 menampilkan tancangan antarmuka halaman detail pantai.



Gambar 4.24 Antarmuka Halaman Detail Pantai

e. Halaman Map dan Rute

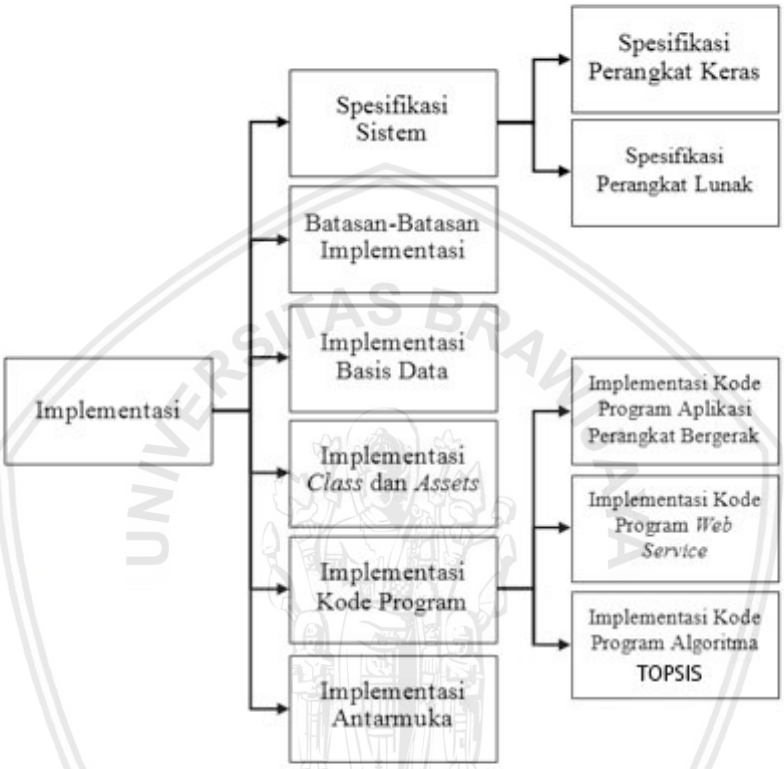
Map dan rute merupakan halaman yang digunakan untuk melihat rute jalan menuju lokasi tempat wisata pantai.. Gambar 4.22 menampilkan rancangan antarmuka halaman map dan rute.



Gambar 4.25 Antarmuka Halaman Map dan Rute

BAB 5 IMPLEMENTASI

Penjelasan terkati bagaimana mengimplementasikan system merupakan isi yang ada pada bab ini. Implementasi ini didasarkan pada hasil dari bab sebelumnya. Gambar 5.1 menjelaskan terkait langkah-langkah dan struktur aplikasi yang dibangun.



Gambar 5.1 Diagram Pohon Implementasi Sistem

Implementasi dibagi menjadi 6 bagian seperti pada Gambar 5.1. Selain itu pada bagian spesifikasi system terdiri dari 2 bagian dan pada implementasi kode program terdiri dari 3 bagian.

5.1 Spesifikasi Sistem

Terdapat 2 jenis spesifikasi system yang menjadi tempat untuk dilakukan impementasi, yaitu.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi computer dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Komponen	Spesifikasi
System Model	Lenovo Legion Y520
Processor	Intel(R) Core(TM) i7-7700HQ CPU 2.80 GHz
Memory	8192 MB DDR4
Display	NVIDIA GeForce GTX 1050

Spesifikasi *smartphone* yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Bergerak

Komponen	Spesifikasi
System Model	Xiaomi Pocophone F1
Processor	Kryo 385 @ 2.8 GHz & 1.8 GHz
Memory	6 GB
Display	Adreno 630

5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi komputer ditunjukkan pada Tabel 5.3.

Tabel 5.3 Spesifikasi Perangkat Lunak pada Komputer

Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Bahasa Pemrograman	Java Android dan PHP + HTML
Tools Development	Android Studio, Sublime Text
Tools Server dan DBMS	XAMPP, MySQL
Web Browser	Mozilla Firefox

Spesifikasi *smartphone* seperti pada Tabel 5.4.

Tabel 5.4 Spesifikasi Perangkat Lunak pada Perangkat Bergerak

Komponen	Spesifikasi
Platform	Android OS 8.1

5.2 Batasan-Batasan Implementasi

Pada tahap implementasi terdapat beberapa batasan dalam proses pengerjaannya, antara lain:

1. Menggunakan Android Studio.

2. Menggunakan bahasa PHP untuk *web service*.
3. *Menggunakan database* MySQL untuk penyimpanan data.

5.3 Implementasi Basis Data

Implementasi penyimpanan data menggunakan *database* MySQL untuk menyimpan data objek wisata pantai di Malang. Perintah SQL yang digunakan untuk implementasi basis data dapat dilihat pada Kode 5.1 sampai Kode 5.7.

```
1. CREATE TABLE IF NOT EXISTS `table_fasilitas` (
2. `id_fasilitas` int(11) NOT NULL PRIMARY KEY,
3. `jenis_fasilitas` varchar(30) NOT NULL,
4. `skor_fasilitas` int(11) NOT NULL )
```

Kode 5.1 Kode Membuat Tabel *table*_fasilitas

Kode 5.1 merupakan kode yang digunakan untuk membuat tabel *table*_fasilitas pada *database*. Terdapat 1 kolom *PRIMARY KEY* dan 2 kolom biasa.

```
1. CREATE TABLE IF NOT EXISTS `table_kriteria` (
2. `id_kriteria` int(11) NOT NULL PRIMARY KEY,
3. `nama_kriteria` varchar(20) NOT NULL,
4. `kode` varchar(20) NOT NULL,
5. `bobot` int(5) NOT NULL,
6. `sifat` varchar(20) NOT NULL )
```

Kode 5.2 Kode Membuat Tabel *table*_kriteria

Kode 5.2 merupakan kode yang digunakan untuk membuat tabel *table*_kriteria di *database*. Terdapat 1 kolom *PRIMARY KEY* dan 4 kolom biasa.

```
1. CREATE TABLE IF NOT EXISTS `table_location` (
2. `id_location` int(11) NOT NULL PRIMARY KEY,
3. `latitude` varchar(16) NOT NULL,
4. `longitude` varchar(64) NOT NULL,
5. `id_pantai` varchar(128) NOT NULL )
```

Kode 5.3 Kode Membuat Tabel *table_location*

Kode 5.3 merupakan kode yang digunakan untuk membuat tabel *table_location* di *database*. Terdapat 1 kolom *PRIMARY KEY* dan 3 kolom biasa.

```
1. CREATE TABLE IF NOT EXISTS `table_pantai` (
2. `id_pantai` int(11) NOT NULL PRIMARY KEY,
3. `nama_pantai` varchar(50) NOT NULL,
4. `biaya_masuk` int(11) NOT NULL,
5. `rating` double NOT NULL,
6. `id_location` int(11) NOT NULL,
7. `image` varchar(30) NOT NULL,
8. `deskripsi` text NOT NULL,
9. `alamat` varchar(100) NOT NULL
10.)
```

Kode 5.4 Kode Membuat Tabel *table*_pantai

Kode 5.4 merupakan kode yang digunakan untuk membuat tabel *table*_pantai di *database*. Terdapat 1 kolom *PRIMARY KEY* dan 7 kolom biasa.

```
1. CREATE TABLE IF NOT EXISTS `table_relasi_fasilitas` (
2. `id_relasi_fasilitas` int(11) NOT NULL PRIMARY KEY,
3. `id_fasilitas` int(11) NOT NULL,
4. `id_pantai` int(11) NOT NULL
5. )
```

Kode 5.5 Kode Membuat Tabel *table*_relasi_fasilitas

Kode 5.5 merupakan kode yang digunakan untuk membuat tabel *table*_relasi_fasilitas di dalam *database*. Terdapat 1 kolom *PRIMARY KEY* dan 2 kolom biasa.

```
1. CREATE TABLE IF NOT EXISTS `table_relasi_transportasi` (
2. `id_relasi_transportasi` int(11) NOT NULL PRIMARY KEY,
3. `id_transportasi` int(11) NOT NULL,
4. `id_pantai` int(11) NOT NULL,
5. )
```

Kode 5.6 Kode Membuat Tabel *table*_relasi_transportasi

Kode 5.6 merupakan kode yang digunakan untuk membuat tabel *table*_relasi_transportasi di *database*. Terdapat 1 kolom *PRIMARY KEY* dan 2 kolom biasa.

```
1. CREATE TABLE IF NOT EXISTS `table_transportasi` (
2. `id_transportasi` int(11) NOT NULL PRIMARY KEY,
3. `jenis_transportasi` varchar(30) NOT NULL,
4. `skor_transportasi` int(11) NOT NULL
5. )
```

Kode 5.7 Kode Membuat Tabel *table*_transportasi

Kode 5.7 merupakan kode yang digunakan untuk membuat tabel *table*_transportasi di *database*. Terdapat 1 kolom *PRIMARY KEY* dan 2 kolom biasa.

Tabel 5.5 sampai Tabel 5.11 merupakan struktur table pada *database* setelah dilakukan implementasi.

Tabel 5.5 Tabel Basis Data *table*_fasilitas

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_fasilitas (PK)	Integer(11)	Untuk menyimpan data id	1,2,3,4, dst
2.	Jenis_fasilitas	Varchar(30)	Untuk menyimpan jenis/nama fasilitas	Rumah Makan
3.	Skor_fasilitas	Integer(11)	Untuk menyimpan nilai skor dari fasilitas	1

Tabel 5.6 Tabel Basis Data *table*_kriteria

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_kriteria (PK)	Integer(11)	Untuk menyimpan data id	1,2,3,4, dst
2.	Nama_kriteria	Varchar(20)	Untuk menyimpan nama kriteria	Biaya masuk
3.	Kode	Varchar(20)	Untuk menyimpan kode dari kriteria	Biaya_masuk
4.	Bobot	Integer(5)	Untuk menyimpan nilai bobot kriteria	1
5.	Sifat	Varchar(20)	Untuk menyimpan data sifat dari kriteria	cost

Tabel 5.7 Tabel Basis Data *table_location*

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_location (PK)	Integer(11)	Untuk menyimpan data id	1,2,3,4, dst
2.	latitude	Double	Untuk menyimpan nilai latitude	-8.348614
3.	Longitude	Double	Untuk menyimpan nilai longitude	112.359036
4.	Id_pantai	Integer(11)	Untuk menyimpan data id dari pantai	1,2,3,4, dst

Tabel 5.8 Tabel Basis Data *table*_pantai

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_pantai(PK)	Integer(11)	Untuk menyimpan data id pantai	1,2,3,4, dst
2.	Nama_pantai	varchar(50)	Untuk menyimpan nama pantai	Pantai Mondangan
3.	Biaya_masuk	Integer(11)	Untuk menyimpan data biaya	5000
4.	Rating	Double	Untuk menyimpan data rating dari pantai	4.3
5.	Id_location	Integer(11)	Untuk menyimpan data id dari lokasi	1,2,3,4, dst

6.	Image	Varchar(30)	Untuk menyimpan data nama gambar	Image-pantai-mondangan
7.	Deskripsi	Text	Untuk menyimpan data deskripsi dari pantai	Pantai yang masih alami....
8.	alamat	Varchar(100)	Untuk menyimpan data alamat dari pantai	Sumberrejo, Sumberoto....

Tabel 5.9 Tabel Basis Data *table*_relasi_fasilitas

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_relasi_fasilitas (PK)	Integer(11)	Untuk menyimpan data id relasi	1,2,3,4, dst
2.	Id_fasilitas	Integer(11)	Untuk menyimpan data id fasilitas	1,2,3,4, dst
3.	Id_pantai	Integer(11)	Untuk menyimpan data id pantai	1,2,3,4, dst

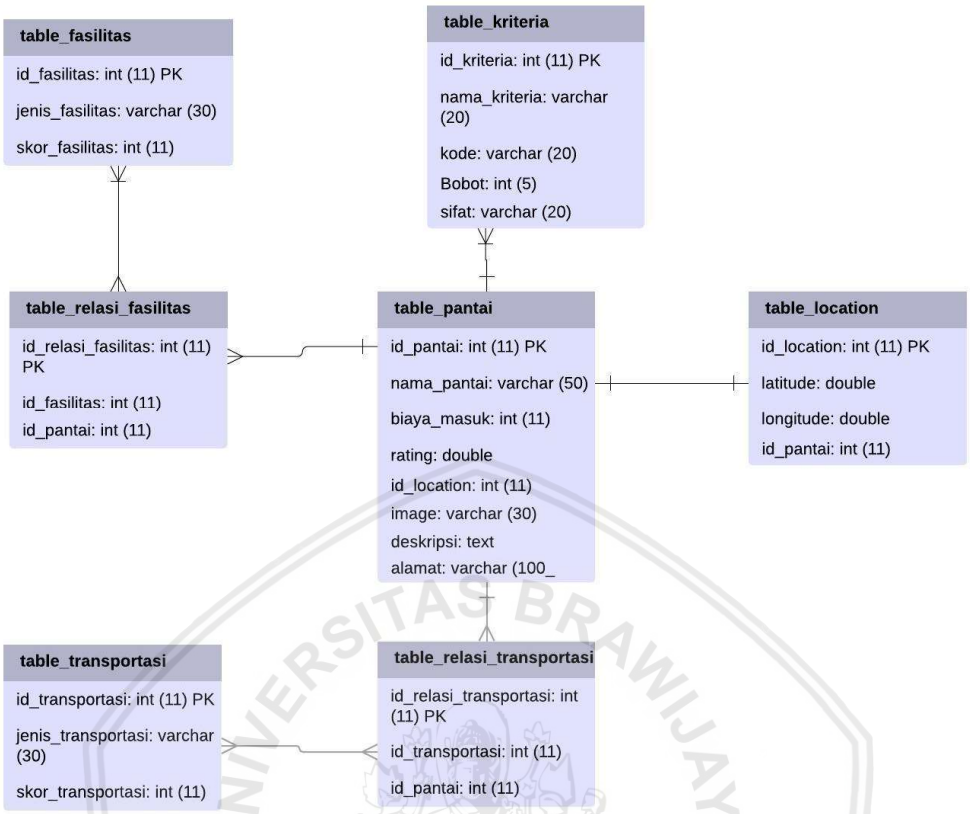
Tabel 5.10 Tabel Basis Data *table*_relasi_transportasi

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_relasi_transportasi (PK)	Integer(11)	Untuk menyimpan data id relasi	1,2,3,4, dst
2.	Id_trasnportasi	Integer(11)	Untuk menyimpan data id transportasi	1,2,3,4, dst
3.	Id_pantai	Integer(11)	Untuk menyimpan data id pantai	1,2,3,4, dst

Tabel 5.11 Tabel Basis Data *table*_transportasi

No.	Nama Kolom	Tipe	Keterangan	Contoh
1.	Id_transportasi (PK)	Integer(11)	Untuk menyimpan data id transportasi	1,2,3,4, dst
2.	Jenis_transportasi	Varchar(30)	Untuk menyimpan jenis transportasi	Motor
3.	Skor_transportasi	Integer(11)	Untuk menyimpan data nilai skor transportasi	1

Gambar 5.2 hasil implementasi sekaligus bentuk relasi antara table yang sudah dibuat.



Gambar 5.2 Hasil Implementasi Basis Data

5.4 Implementasi *Class* dan *Assets* pada File Program

Impelementasi ini merupakan implementasi yang mengacu pada perancangan class diagram di bab sebelumnya. Tabel 5.12 merupakan hasil dari implementasi perancangan tersebut.

Tabel 5.12 Berkas Kelas dan *Assets Layout* pada Aplikasi Perangkat Bergerak

No.	*Package*	*Folder*	Nama Kelas	Nama *Layout*
1.	app.com.example.hilmi.seatraveler	Activity	SplashActivity.java	splashscreen.xml
2.	app.com.example.hilmi.seatraveler	Activity	HomeActivity.java	activity_home.xml
3.	app.com.example.hilmi.seatraveler	Activity	MintaRekomendasiActivity.java	activity_mitan_rekomendasi.xml
4.	app.com.example.hilmi.seatraveler	Activity	HasilRekomendasiActivity.java	activity_hasil_rekomendasi.xml
5.	app.com.example.hilmi.seatraveler	Activity	DetailPantaiActivity.java	activity_detail_pantai.xml
6.	app.com.example.hilmi.seatraveler	-	OutScreenBottomSheetBehavior.java	-

7.	app.com.example.hilmi.seatraveler	Adapter	Adapter_pantai.java	-
8.	app.com.example.hilmi.seatraveler	Model	pantai.java	-

5.5 Implementasi Kode Program

Pada implementasi ini menjelaskan implementasi tiap-tiap kelas ke dalam bentuk kode program, sesuai dengan yang tertera pada Tabel 5.12 dan mengacu pada perancangan yang ada pada bab sebelumnya. Terdapat pula implementasi kode program untuk *web service*. Implementasi algoritma sebagai metode rekomendasi objek wisata pantai di Malang yang mengacu pada perancangan algoritma dalam Gambar 4.19.

5.5.1 Implementasi Kode Program Aplikasi Perangkat Bergerak

Kode 5.8 merupakan hasil implementasi kelas MintaRekomendasiActivity yang dimana terdapat kode yang digunakan untuk menggunakan sensor GPS untuk mengetahui posisi *latitude* dan *longitude smartphone*. Sementara Kode 5.9 merupakan hasil implementasi kelas HasilRekomendasiActivity yang dimana terdapat fungsi untuk mendapatkan hasil rekomendasi dari *webb service.*

```
1. public class MintaRekomendasiActivity extends AppCompatActivity {
2.
3.    private Button proses;
4.    private EditText jarak, htm, rating, fasilitas, transportasi;
5.    private ImageButton backButton;
6.    public ConnectivityManager conMgr;
7.    private FusedLocationProviderClient myLocation;
8.    private static final int LOCATION_REQUEST = 500;
9.    private String latitude, longitude;
10.
11.   protected void onCreate(Bundle savedInstanceState) {
12.     super.onCreate(savedInstanceState);
13.     setContentView(R.layout.activity_minta_rekomendasi);
14.
15.     conMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
16.
17.     //get latitude and longitude from GPS
18.     myLocation =
    LocationServices.getFusedLocationProviderClient(MintaRekomendasiActiv
    ity.this);
19.
20.     if
    (ActivityCompat.checkSelfPermission(MintaRekomendasiActivity.this,
    android.Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED &&
    ActivityCompat.checkSelfPermission(MintaRekomendasiActivity.this,
    android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED) {
21.       ActivityCompat.requestPermissions(MintaRekomendasiActivity.this
    , new String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},
    LOCATION_REQUEST);
```

```
22.       return;
23.     }
24.
25.     myLocation.getLastLocation().addOnSuccessListener(MintaRekomendasiAct
    ivity.this, new OnSuccessListener<Location>() {
26.       @Override
27.       public void onSuccess(Location location) {
28.         latitude = String.valueOf(location.getLatitude());
29.         longitude = String.valueOf(location.getLongitude());
30.       }
31.     });
32.
33.     jarak = findViewById(R.id.editTextJarak);
34.     rating = findViewById(R.id.editTextRating);
35.     htm = findViewById(R.id.editTextHTM);
36.     fasilitas = findViewById(R.id.editTextFasilitas);
37.     transportasi = findViewById(R.id.editTextTransportasi);
38.     proses = findViewById(R.id.bottom_button);
39.     proses.setOnClickListener(new View.OnClickListener() {
40.       @Override
41.       public void onClick(View view) {
42.         String valueJarak = jarak.getText().toString();
43.         String valueHTM = htm.getText().toString();
44.         String valueRating = rating.getText().toString();
45.         String valueFasilitas = fasilitas.getText().toString();
46.         String valueTransportasi =
    transportasi.getText().toString();
47.
48.         // mengecek kolom yang kosong dan internet
49.         if (valueJarak.trim().length() > 0 &&
    valueHTM.trim().length() > 0 && valueRating.trim().length() > 0 &&
    valueFasilitas.trim().length() > 0 &&
    valueTransportasi.trim().length() > 0) {
50.           if (conMgr.getActiveNetworkInfo() != null &&
    conMgr.getActiveNetworkInfo().isAvailable() &&
    conMgr.getActiveNetworkInfo().isConnected()) {
51.             Intent intent = new
    Intent(MintaRekomendasiActivity.this,
    HasilRekomendasiActivity.class);
52.             intent.putExtra("valueJarak", valueJarak);
53.             intent.putExtra("valueHTM", valueHTM);
54.             intent.putExtra("valueRating", valueRating);
55.             intent.putExtra("valueFasilitas", valueFasilitas);
56.             intent.putExtra("valueTransportasi",
    valueTransportasi);
57.             intent.putExtra("latitude", latitude);
58.             intent.putExtra("longitude", longitude);
59.             startActivity(intent);
60.           } else {
61.             Toast.makeText(getApplicationContext() ,"No Internet
    Connection", Toast.LENGTH_LONG).show();
62.           }
63.         } else {
64.           Toast.makeText(getApplicationContext() ,"Kolom tidak
    boleh kosong", Toast.LENGTH_LONG).show();
65.         }
66.       }
67.     });
```

```
68.
69.        backButton = findViewById(R.id.backButton);
70.        backButton.setOnClickListener(new View.OnClickListener()
    {
71.          @Override
72.          public void onClick(View view) {
73.            finish();
74.          }
75.        });
76.      }
77.    }
78.
```

Kode 5.8 Kode Program Kelas MintaRekomendasiActivity

Kode 5.8 menampilkan implementasi kode program kelas untuk mengakses sensor GPS dan mendapatkan nilai posisi latitude dan longitude dari *smartphone*. Penjelasan potongan kode program kelas MintaRekomendasiActivity, yaitu:

1. Baris 3-9, merupakan perintah deklarasi *variable* global.
2. Baris 11-77, merupakan *method onCreate* dimana pada *method* ini terdapat inisialisasi *variable* global, memberikan perintah *clickable* pada *button* dan juga terdapat perintah untuk mengakses sensor GPS.
3. Baris 17-31, merupakan perintah untuk mengakses sensor GPS menggunakan kelas *FusedLocationProviderClient*.
4. Baris 25-31, merupakan perintah untuk mendapatkan nilai latitude dan longitude, lalu memasukan nilai tersebut kedalam variabel.
5. Baris 33-39, merupakan perintah untuk menginisialisasi variabel.
6. Baris 39-67, merupakan perintah memproses perintah ketika menekan tombol minta rekomendasi, terdapat proses *intent* untuk mengirim data kepada HasilRekomendasiActivity.
7. Baris 69-76, merupakan perintah untuk menutup activity ketika menekan tombol kembali.

```
1.  public class HasilRekomendasiActivity extends AppCompatActivity {
2.      private RecyclerView recyclerView;
3.      private Adapter_pantai adapter;
4.      private ArrayList<pantai> list_data = new ArrayList<>();
5.      private LinearLayoutManager layoutManager;
6.      private ImageButton backButton;
7.      private String latitude, longitude, valueJarak, valueHTM, valueRating, valueFasilitas,
    valueTransportasi;
8.
9.      @Override
10.     protected void onCreate(Bundle savedInstanceState) {
11.             super.onCreate(savedInstanceState);
12.             setContentView(R.layout.activity_hasil_rekomendasi);
13.
14.             //get data from intent
15.             Intent intent = getIntent();
16.             latitude=intent.getExtras().getString("latitude");
17.             longitude=intent.getExtras().getString("longitude");
18.             valueJarak=intent.getExtras().getString("valueJarak");
19.             valueHTM=intent.getExtras().getString("valueHTM");
```

```
20.             valueRating=intent.getExtras().getString("valueRating");
21.             valueFasilitas=intent.getExtras().getString("valueFasilitas");
22.             valueTransportasi=intent.getExtras().getString("valueTransportasi");
23.
24.             //calling method getData for get JSON rekomendasi
25.             getData();
26.             //code for recycler view that showing result of recommendation
27.             recyclerView = (RecyclerView) findViewById(R.id.recycler);
28.             layoutManager = new LinearLayoutManager(HasilRekomendasiActivity.this,
    LinearLayoutManager.VERTICAL, false);
29.             adapter = new Adapter_pantai(HasilRekomendasiActivity.this,this.list_data);
30.             recyclerView.setLayoutManager(layoutManager);
31.             recyclerView.setHasFixedSize(false);
32.             recyclerView.setAdapter(adapter);
33.
34.             //press back button
35.             backButton = findViewById(R.id.backButton);
36.             backButton.setOnClickListener(new View.OnClickListener() {
37.                     @Override
38.                     public void onClick(View view) {
39.                             finish();
40.                     }
41.             });
42.             }
43.
44.     public void getData(){
45.             final ProgressDialog progressDialog = new
    ProgressDialog(HasilRekomendasiActivity.this);
46.             progressDialog.setMessage("Loading...");
47.             progressDialog.show();
48.             RequestQueue requestQueue =
    Volley.newRequestQueue(HasilRekomendasiActivity.this);
49.             String domain = getResources().getString(R.string.url);
50.             String url = domain+"pantai_gps.php";
51.             StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new
    Response.Listener<String>() {
52.                     }
53.
54.                     map.setOnMapLoadedCallback(new GoogleMap.OnMapLoadedC
    allback() {
55.                     @Override
56.                     public void onResponse(String response) {
57.                     try {
58.                     Log.e("tagconvertstr", "["+response+"]");
59.                     JSONObject jsonObject = new JSONObject(response);
60.                     JSONArray jsonArray = jsonObject.getJSONArray("rekomendasi_pantai");
61.                     for (int a = 0; a < jsonArray.length(); a ++){
62.                         JSONObject json = jsonArray.getJSONObject(a);
63.                         pantai model = new pantai();
64.                         model.setId_pantai(json.getString("id_pantai"));
65.                         model.setNama_pantai(json.getString("nama_pantai"));
66.                         model.setBiaya_masuk(json.getString("biaya_masukNonTOPSIS"));
67.                         model.setRating(json.getString("ratingNonTOPSIS"));
68.                         model.setTransportasi(json.getString("transportasiNonTOPSIS"));
69.                         model.setJarak(json.getString("jarakNonTOPSIS"));
70.                         model.setDplus(json.getString("Dplus"));
71.                         model.setDminus(json.getString("Dminus"));
```

```
72.                             model.setJarak_solusi(json.getString("jarak_solusi"));
73.                             model.setLatitude(json.getString("latitude"));
74.                             model.setLongitude(json.getString("longitude"));
75.                             model.setImage(json.getString("image"));
76.                             model.setDeskripsi(json.getString("deskripsi"));
77.                             model.setAlamat(json.getString("alamat"));
78.                             model.setMyLatitude(latitude);
79.                             model.setMyLongitude(longitude);
80.                             list_data.add(model);
81.                         }
82.                         } catch (JSONException e) {
83.                             e.printStackTrace();
84.                             progressDialog.dismiss();
85.                         }
86.                         adapter.notifyDataSetChanged();
87.                         progressDialog.dismiss();
88.                         }
89.                 }, new Response.ErrorListener() {
90.
91.                         @Override
92.                         public void onErrorResponse(VolleyError error) {
93.                         Toast.makeText(HasilRekomendasiActivity.this, error.getMessage(),
    Toast.LENGTH_SHORT).show();
94.                         Log.e("Error valley","Volley");
95.                         progressDialog.dismiss();
96.                 }
97.                 }) {
98.                         @Override
99.                         protected Map<String, String> getParams() {
100.                            // Posting parameters
101.                            Map<String, String> params = new HashMap<String, String>();
102.                            params.put("latitude", latitude);
103.                            params.put("longitude", longitude);
104.                            params.put("valueJarak", valueJarak);
105.                            params.put("valueHTM", valueHTM);
106.                            params.put("valueRating", valueRating);
107.                            params.put("valueFasilitas", valueFasilitas);
108.                            params.put("valueTransportasi", valueTransportasi);
109.
110.                            return params;
111.                        }
112.                };
113.
114.                requestQueue.add(stringRequest);
115.    }
116.  }
```

Kode 5.9 Kode Program Kelas HasilRekomendasiActivity

Kode 5.9 merupakan implementasi kode untuk mendapatkan hasil rekomendasi dari *web service*. Penjelasan kode program kelas HasilRekomendasiActivity untuk mendapatkan hasil rekomendasi, yaitu:

1. Baris 2-7, merupakan perintah deklarasi *variable* global.
2. Baris 10-42, merupakan *method onCreate* dimana pada *method* ini terdapat inisialisasi *variable* global, memberikan perintah *clickable* pada *button* dan juga terdapat perintah untuk mengekseksi *method* lain.

3. Baris 25, merupakan perintah untuk mengeksekusi *method* getData().
4. Baris 44-115, merupakan *method* getData() dimana terdapat perintah untuk mengirim data ke *web service* dan juga mengambil data hasil rekomendasi dari *web servicesss.*
5. Baris 50-81, merupakan perintah untuk mengambil data hasil rekomendasi dari *web service*.
6. Baris 61-80, merupakan perintah memodelkan data json yang sudah didapat dari *web service*.
7. Baris 99-111, merupakan perintah mengirim data parameter kepada *web service*.

5.5.2 Implementasi Kode Program *Web Service*

Implementasi ini menjelaskan perancangan akses data pada bab sebelumnya dan hanya dicantumkan kode program dari fungsi-fungsi untuk mendapatkan hasil rekomendasi berformat JSON yang mengacu dalam Gambar 4.10 sampai Gambar 4.11.

```
1.    $query = mysqli_query($con, "SELECT nama_pantai FROM
   `table_pantai`");
2.    $json = '{"daftar_pantai": [';
3.    $no = 1;
4.
5.    while ($row = mysqli_fetch_array($query)){
6.          $char ='"';
7.          $json .=
8.          '{
9.                "nama_pantai":"'.$no.'.
   '.str_replace($char,'`',strip_tags($row['nama_pantai'])).'"
10.         },';
11.         $no++;
12.   }
13.   // buat menghilangkan koma diakhir array
14.   $json = substr($json,0,strlen($json)-1);
15.
16.   $json .= ']}';
17.
18.   // print json
19.   echo $json;
20.
21.   mysqli_close($con);
22.
23.
```

Kode 5.10 Kode Program Fungsi Mendapatkan Daftar Pantai

Kode 5.9 menampilkan implementasi kode program dari potongan fungsi *web service* untuk mendapatkan daftar pantai. Penjelasan implementasi kode program, yakni:

1. Baris 1, perintah untuk mengambil data dari database.
2. Baris 2, variabel json yang menyimpan data daftar pantai dengan format penulisan json.

3. Baris 5-12, merupakan perintah memasukkan data yang didapat dari database kedalam variabel json.
4. Baris 19, merupakan perintah untuk mengprint nilai dari variabel json.

```
1.     $latGPS = trim($_POST["latitude"]);
2.     $longGPS = trim($_POST["longitude"]);
3.     $HTM = trim($_POST["valueHTM"]);
4.     $Jarak = trim($_POST["valueJarak"]);
5.     $Rating = trim($_POST["valueRating"]);
6.     $Transportasi = trim($_POST["valueTransportasi"]);
7.     $Fasilitas = trim($_POST["valueFasilitas"]);
8.
9.     $query = mysqli_query($con, "SELECT table_pantai.id_pantai,
   table_pantai.nama_pantai, table_pantai.biaya_masuk,
   table_pantai.rating, table_location.latitude,
   table_location.longitude, table_pantai.image, table_pantai.deskripsi,
   table_pantai.alamat FROM `table_pantai` INNER JOIN table_location ON
   table_pantai.id_location = table_location.id_location");
10.
11.    $query_kriteria = mysqli_query($con, "SELECT * FROM
   `table_kriteria`");
12.    $kriteria = $kriteria_seq = array();
13.    while ($row = mysqli_fetch_array($query_kriteria)){
14.      $kriteria[$row['kode']] = $row;
15.      $kriteria_seq[] = $row;
16.    }
17.
18.    $data_awal_penilaian = array();
19.    while ($row = mysqli_fetch_array($query)){
20.      $query_jumlah_transportasi = mysqli_query($con, "SELECT
   COUNT(jenis_transportasi) AS transportasi FROM `table_transportasi`
   INNER JOIN table_relasi_transportasi ON
   table_relasi_transportasi.id_transportasi =
   table_transportasi.id_transportasi INNER JOIN table_pantai ON
   table_relasi_transportasi.id_pantai = table_pantai.id_pantai WHERE
   table_pantai.id_pantai = ".$row['id_pantai']);
21.     $query_jumlah_fasilitas = mysqli_query($con, "SELECT
   COUNT(jenis_fasilitas) AS fasilitas FROM `table_fasilitas` INNER JOIN
   table_relasi_fasilitas ON table_relasi_fasilitas.id_fasilitas =
   table_fasilitas.id_fasilitas INNER JOIN table_pantai ON
   table_relasi_fasilitas.id_pantai = table_pantai.id_pantai WHERE
   table_pantai.id_pantai = ".$row['id_pantai']);
22.     $query_transportasi = mysqli_query($con, "SELECT
   (jenis_transportasi) AS transportasi FROM `table_transportasi` INNER
   JOIN table_relasi_transportasi ON
   table_relasi_transportasi.id_transportasi =
   table_transportasi.id_transportasi INNER JOIN table_pantai ON
   table_relasi_transportasi.id_pantai = table_pantai.id_pantai WHERE
   table_pantai.id_pantai = ".$row['id_pantai']);
23.     $query_fasilitas = mysqli_query($con, "SELECT (jenis_fasilitas)
   AS fasilitas FROM `table_fasilitas` INNER JOIN table_relasi_fasilitas
   ON table_relasi_fasilitas.id_fasilitas = table_fasilitas.id_fasilitas
   INNER JOIN table_pantai ON table_relasi_fasilitas.id_pantai =
   table_pantai.id_pantai WHERE table_pantai.id_pantai =
   ".$row['id_pantai']);
24.     $jumlah_transportasi =
   mysqli_fetch_array($query_jumlah_transportasi);
25.     $jumlah_fasilitas = mysqli_fetch_array($query_jumlah_fasilitas);
26.     $transportasi = "";
27.     $fasilitas = "";
28.     while ($row_transportasi =
   mysqli_fetch_array($query_transportasi)) {
```


repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
29.       $transportasi .= $row_transportasi['transportasi'].", ";
30.     }
31.     while ($row_fasilitas = mysqli_fetch_array($query_fasilitas)) {
32.       $fasilitas .= $row_fasilitas['fasilitas'].", ";
33.     }
34.     $jarak = GetDrivingDistance($latGPS, $row['latitude'] ,$longGPS,
    $row['longitude']);
35.
36.     $data = array(
37.                 "id_pantai"=>$row['id_pantai'],
38.                 "nama_pantai"=>$row['nama_pantai'],
39.                 "biaya_masuk"=>$row['biaya_masuk'],
40.                 "rating"=>$row['rating'],
41.                 "transportasi"=>$jumlah_transportasi['transportasi'],
42.                 "transportasiNonTOPSIS"=>$transportasi,
43.                 "fasilitas"=>$jumlah_fasilitas['fasilitas'],
44.                 "fasilitasNonTOPSIS"=>$fasilitas,
45.                 "jarak"=>$jarak['distance'],
46.                 "latitude"=>$row['latitude'],
47.                 "longitude"=>$row['longitude'],
48.                 "image"=>$row['image'],
49.                 "deskripsi"=>$row['deskripsi'],
50.                 "alamat"=>$row['alamat']
51.      );
52.      $data_awal_penilaian[] = $data;
53.      }
54.
55.    $data_ternormalisasi =
    getKeputusanTernormalisasi($data_awal_penilaian);
56.
57.   $data_terbobot =
    getKeputusanTerbobot(getKeputusanTernormalisasi($data_awal_penilaian)
    , $Jarak, $HTM, $Rating, $Fasilitas, $Transportasi);
58.
59.   $matriks_ideal_positif = getMatriksIdealPositif($data_terbobot,
    $kriteria_seq);
60.
61.   $matriks_ideal_negatif = getMatriksIdealNegatif($data_terbobot,
    $kriteria_seq);
62.
63.   $matriks_solusi = getJarakSolusiIdeal($data_terbobot,
    $matriks_ideal_positif, $matriks_ideal_negatif, $kriteria_seq);
64.
65.   $Ranking = getRanking($matriks_solusi);
66.
67.   $json = '{"rekomendasi_pantai": [';
68.
69.   for($i=0; $i<count($matriks_solusi);$i++){
70.
71.    $char ='"';
72.    $json .=
73.      '{
74.        "id_pantai":"'.str_replace($char,'`',strip_tags($Ranking[$i]['
    id_pantai'])).'",
75.        "nama_pantai":"'.str_replace($char,'`',strip_tags($Ranking[$i]
    ['nama_pantai'])).'",
76.        "biaya_masuk":"'.str_replace($char,'`',strip_tags($Ranking[$i]
    ['biaya_masuk'])).'",
77.        "biaya_masukNonTOPSIS":"'.str_replace($char,'`',strip_tags($Ra
    nking[$i]['biaya_masukNonTOPSIS'])).'",
78.        "rating":"'.str_replace($char,'`',strip_tags($Ranking[$i]['rat
    ing'])).'",
79.        "ratingNonTOPSIS":"'.str_replace($char,'`',strip_tags($Ranking
    [$i]['ratingNonTOPSIS'])).'",
```

```
80.         "transportasi":"'.str_replace($char,'``',strip_tags($Ranking[$i
    ]['transportasi'])).'",
81.         "transportasiNonTOPSIS":"'.str_replace($char,'``',strip_tags($R
    anking[$i]['transportasiNonTOPSIS'])).'",
82.         "fasilitas":"'.str_replace($char,'``',strip_tags($Ranking[$i]['
    fasilitas'])).'",
83.         "fasilitasNonTOPSIS":"'.str_replace($char,'``',strip_tags($Rank
    ing[$i]['fasilitasNonTOPSIS'])).'",
84.         "latitude":"'.str_replace($char,'``',strip_tags($Ranking[$i]['l
    atitude'])).'",
85.         "longitude":"'.str_replace($char,'``',strip_tags($Ranking[$i]['
    longitude'])).'",
86.         "jarak":"'.str_replace($char,'``',strip_tags($Ranking[$i]['jara
    k'])).'",
87.         "jarakNonTOPSIS":"'.str_replace($char,'``',strip_tags($Ranking[
    $i]['jarakNonTOPSIS'])).'",
88.         "Dplus":"'.str_replace($char,'``',strip_tags($Ranking[$i]['D+']
    )).'",
89.         "Dminus":"'.str_replace($char,'``',strip_tags($Ranking[$i]['D-
    ']).'",
90.         "jarak_solusi":"'.str_replace($char,'``',strip_tags($Ranking[$i
    ]['jarak_solusi'])).'",
91.         "image":"'.str_replace($char,'``',strip_tags($Ranking[$i]['imag
    e'])).'",
92.         "deskripsi":"'.str_replace($char,'``',strip_tags($Ranking[$i]['
    deskripsi'])).'",
93.         "alamat":"'.str_replace($char,'``',strip_tags($Ranking[$i]['ala
    mat'])).'"
94.       },';
95.
96.     // buat menghilangkan koma diakhir array
97.     $json = substr($json,0,strlen($json)-1);
98.
99.     $json .= ']}';
100.
101.    // print json
102.    echo $json;
103.
104.    mysqli_close($con);
105.
106.
```

Kode 5.11 Kode Program Fungsi Mendapatkan Hasil Rekomendasi Pantai

Kode 5.10 menampilkan fungsi mendapatkan hasil rekomendasi pantai ke dalam format JSON. Penjelasan implementasi kode program, yaitu:

1. Baris 1-7, merupakan perintah untuk memasukkan data parameter yang didapat dari aplikasi kedalam variabel.
2. Baris 9-52, merupakan perintah untuk mengambil data dari database dan menaruh data yang didapat kedalam variabel.
3. Baris 34, merupakan perintah untuk mendapatkan nilai jarak lokasi dan memasukkannya kedalam variabel.
4. Baris 55, merupakan perintah untuk mendapatkan nilai data ternormalisasi.
5. Baris 57, merupakan perintah untuk mendapatkan nilai data terbobot.
6. Baris 59, merupakan perintah untuk mendapatkan nilai matriks ideal positif.
7. Baris 61, merupakan perintah untuk mendapatkan nilai matriks ideal negative.
8. Baris 63, merupakan perintah untuk mendapatkan nilai matriks solusi.
9. Baris 65, merupakan perintah untuk mendapatkan hasil perangkingan.

10. Baris 67-102, merupakan perintah untuk memasukkan nilai perangkingan kedalam variabel json dengan format penulisan json.

5.5.3 Implementasi Kode Program Algoritma TOPSIS

Algoritma TOPSIS digunakan untuk menghitung rekomendasi tempat wisata pantai. Pada implementasi ini dicantumkan kode program algoritma TOPSIS yang mengacu pada perancangan TOPSIS dalam Gambar 4.15. Implementasi algoritma yang dicantumkan berupa fungsi menghitung jarak alternatif terhadap solusi ideal dan nilai preferensi.

```
1.    function getJarakSolusiIdeal($dataTerbobot, $idealPositif,
   $idealNegatif, $kriteria_seq)
2.    {
3.      $matriks_jarak_solusi = array();
4.      $jarak_positif = array();
5.      $jarak_negatif = array();
6.      for($i=0; $i<count($dataTerbobot);$i++){
7.        $jarak_positif[] = 0;
8.        $jarak_negatif[] = 0;
9.
10.       for($y=0; $y<count($kriteria_seq);$y++){
11.         $jarak_positif[$i] +=
   pow(($idealPositif[$kriteria_seq[$y]['kode']]-
   $dataTerbobot[$i][$kriteria_seq[$y]['kode']]), 2);
12.         $jarak_negatif[$i] +=
   pow(($dataTerbobot[$i][$kriteria_seq[$y]['kode']]-
   $idealNegatif[$kriteria_seq[$y]['kode']]), 2);
13.       }
14.
15.       $jarak_positif[$i] = sqrt($jarak_positif[$i]);
16.       $jarak_negatif[$i] = sqrt($jarak_negatif[$i]);
17.
18.       $data = array(
19.                 "id_pantai"=>$dataTerbobot[$i]['id_pantai'],
20.                 "nama_pantai"=>$dataTerbobot[$i]['nama_pantai'],
21.                 "biaya_masuk"=>$dataTerbobot[$i]['biaya_masuk'],
22.
     "biaya_masukNonTOPSIS"=>$dataTerbobot[$i]['biaya_masukNonTOPSIS'],
23.                 "rating"=>$dataTerbobot[$i]['rating'],
24.
     "ratingNonTOPSIS"=>$dataTerbobot[$i]['ratingNonTOPSIS'],
25.                 "transportasi"=>$dataTerbobot[$i]['transportasi'],
26.
     "transportasiNonTOPSIS"=>$dataTerbobot[$i]['transportasiNonTOPSIS'
   ],
27.                 "fasilitas"=>$dataTerbobot[$i]['fasilitas'],
28.
     "fasilitasNonTOPSIS"=>$dataTerbobot[$i]['fasilitasNonTOPSIS'],
29.                 "latitude"=>$dataTerbobot[$i]['latitude'],
30.                 "longitude"=>$dataTerbobot[$i]['longitude'],
31.                 "jarak"=>$dataTerbobot[$i]['jarak'],
32.
     "jarakNonTOPSIS"=>$dataTerbobot[$i]['jarakNonTOPSIS'],
33.                 "D+"=>$jarak_positif[$i],
34.                 "D-"=>$jarak_negatif[$i],
35.
     "jarak_solusi"=>$jarak_negatif[$i]/($jarak_negatif[$i]+$jarak_posi
   tif[$i]),
36.                 "image"=>$dataTerbobot[$i]['image'],
37.                 "deskripsi"=>$dataTerbobot[$i]['deskripsi'],
38.                 "alamat"=>$dataTerbobot[$i]['alamat']
```

```
39.       );
40.
41.       $matriks_jarak_solusi[] = $data;
42.     }
43.     return $matriks_jarak_solusi;
44. }
```

Kode 5.12 Kode Program Fungsi Menghitung Jarak Alternatif Terhadap Solusi Ideal dan Nilai Prefensi

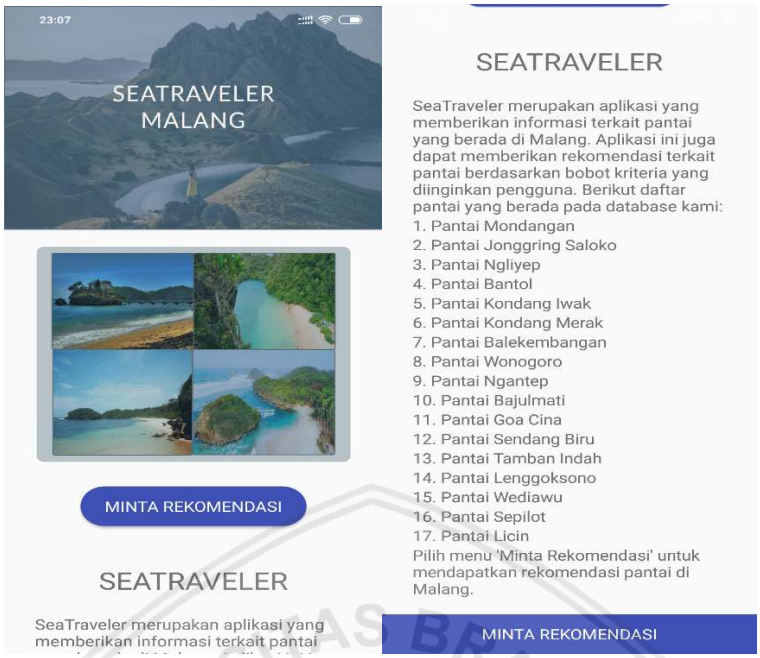
Kode 5.12 menampilkan implementasi kode program dari algoritma TOPSIS untuk fungsi menghitung jarak alternative terhadap solusi ideal. Penjelasan implementasi kode program, yaitu:

1. Baris 1, merupakan nama metode untuk mendapatkan nilai jarak dengan parameter berupa data nilai terbobot, nilai matriks ideal positif, nilai matriks ideal negatif dan data kriteria.
2. Baris 11, merupakan perintah menghitung nilai matriks ideal positif dikurangi nilai data terbobot lalu dikuadratkan.
3. Baris 12, merupakan perintah menghitung nilai data terbobot dikurangi matriks ideal lalu dikuadratkan.
4. Baris 15-16, merupakan perintah untuk menghitung akar kuadrat dari nilai hasil baris 11 dan 12.
5. Baris 18-41, merupakan perintah untuk memasukkan nilai kedalam array.
6. Baris 35, merupakan perintah untuk menghitung jarak solusi.

5.6 Implementasi Antarmuka

Hasil dari implementasi antarmuka yang mengacu pada perancangan antarmuka pada bab sebelumnya dapat dilihat pada Gambar 5.3 sampai Gambar 5.6.

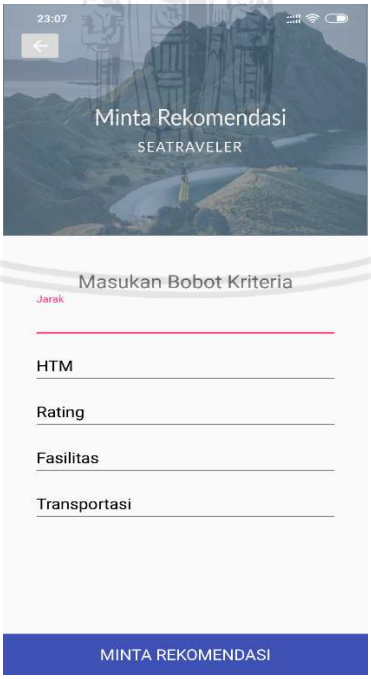
a. Antarmuka *Home*



Gambar 5.3 Hasil Implementasi Antarmuka *Home*

b. Antarmuka *Input* Bobot Kriteria

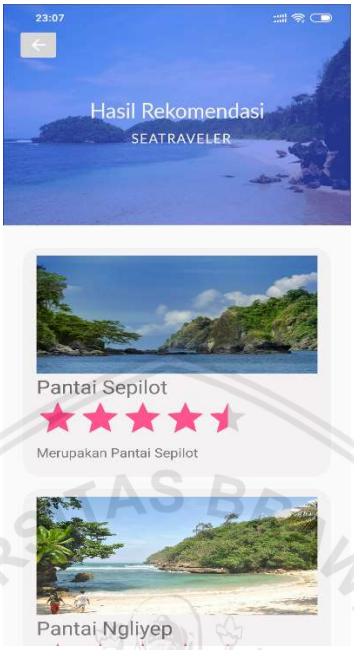
Input bobot kriteria merupakan halaman yang digunakan untuk memasukkan nilai bobot kriteria. Halaman *Input* bobot kriteria menampilkan form isian yaitu jarak, htm, rating, fasilitas dan transportasi. Implementasi halaman *input* bobot kriteria dapat dilihat dalam Gambar 5.4.



Gambar 5.4 Hasil Implementasi Antarmuka *Input* Bobot Kriteria

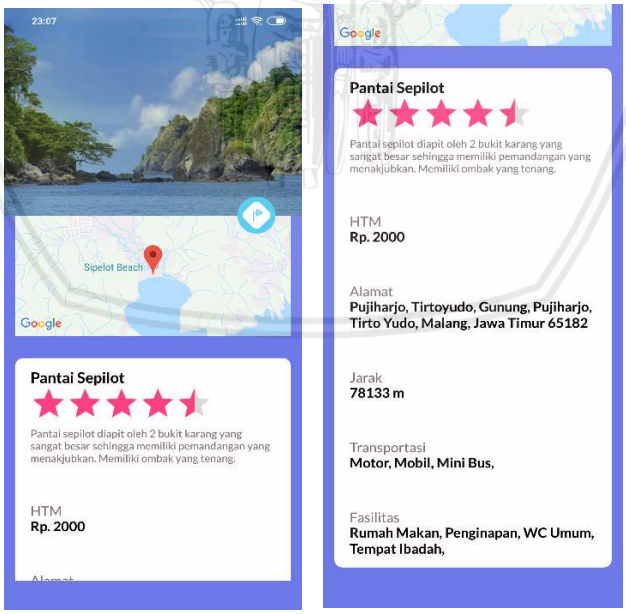
c. Antarmuka Hasil Rekomendasi

Implementasi antarmuka hasil rekomendasi dapat dilihat dalam Gambar 5.5. Hasil rekomendasi merupakan halaman yang digunakan untuk melihat hasil rekomendasi tempat wisata pantai.



Gambar 5.5 Hasil Implementasi Antarmuka Hasil Rekomendasi

d. Antarmuka Detail Pantai

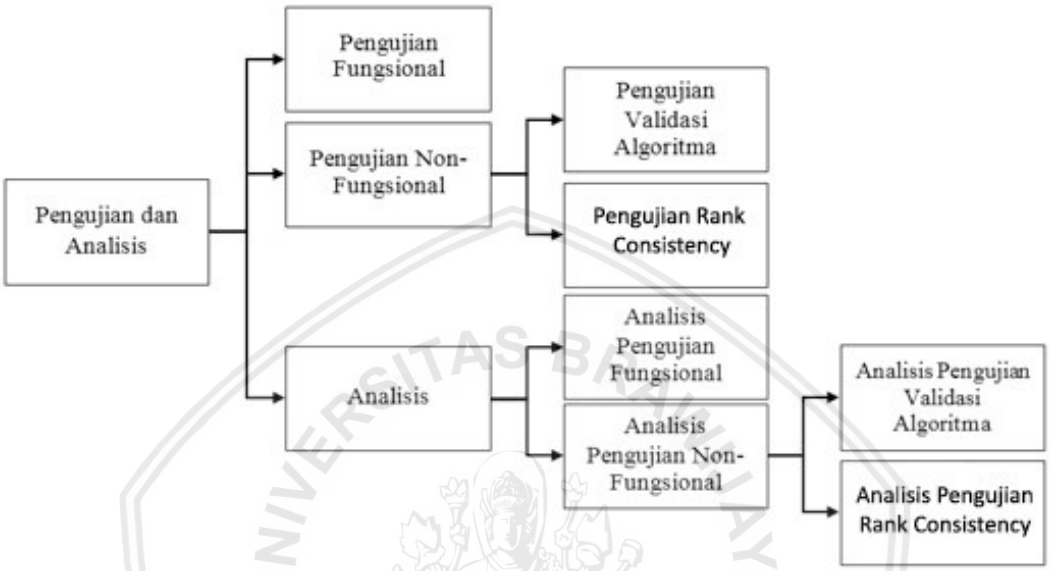


Gambar 5.6 Hasil Implementasi Antarmuka Detail Pantai

Implementasi antarmuka detail pantai dapat dilihat dalam Gambar 5.6. Detail pantai merupakan halaman yang digunakan untuk melihat informasi pantai yang dipilih.

BAB 6 PENGUJIAN DAN ANALISIS

Merupakan proses untuk melakukan pengujian dan analisis dari hasil implementasi yang sudah dilakukan pada bab sebelumnya. Penjelasan terkait struktur pengujian dan analisis sistem rekomendasi tempat wisata pantai terdapat pada Gambar 6.1.



Gambar 6.1 Pohon Diagram Pengujian dan Analisis

Pada bab ini terdapat 3 bagian, yakni: pengujian fungsional, pengujian non fungsional dan analisis.

6.1 Pengujian Fungsional

Digunakan untuk mengetahui apakah system yang telah dibuat sudah memenuhi segala kebutuhan fungsionalnya. Pengujian fungsionalitas pada penelitian kali ini menggunakan metode *blackbox.* Acuan pada pengujian ini terdapat pada *scenario usecase* yang telah dibuat sebelumnya.

Pada pengujian ini, pengujian hanya dilakukan pada *scenario usecase* dengan aliran utama saja. Hal ini dikarenakan sesuai dengan Batasan masalah yang sudah disebutkan diatas, yakni kondisi GPS dan internet sudah dianggap dalam kondisi yang baik. Pengujian fungsional dapat dilihat pada Tabel 6.1 sampai Tabel 6.5.

Tabel 6.1 Kasus Uji Fungsional Menampilkan Daftar Pantai

Nomor Kasus Uji	**KUF-001**
Nama Kasus Uji	**Menampilkan Daftar Pantai**
Objek Uji	Kebutuhan Fungsional

Tujuan Pengujian	Memastikan bahwa system dapat memenuhi kebutuhan dalam menampilkan daftar pantai yang terdapat pada *database.*
Prosedur Pengujian	1. Buka aplikasi. 2. Masuk ke halaman *home*.
Hasil yang Diharapkan	Pengguna dapat membuka aplikasi dan masuk ke halaman *home.* Pengguna juga dapat melihat daftar pantai pada halaman *home*

Tabel 6.1 merupakan prosedur kasus uji fungsional menampilkan daftar pantai. Detail kasus uji memasukkan alternatif dan kriteria terdapat pada Lampiran 1.

Tabel 6.2 Kasus Uji Fungsional Memasukkan Bobot Kriteria

Nomor Kasus Uji	**KUF-002**
Nama Kasus Uji	**Memasukkan Bobot Kriteria**
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan form *input,* sehingga pengguna dapat memasukkan bobot kriteria.
Prosedur Pengujian	1. Buka aplikasi. 2. Masuk ke halaman *home.* 3. Pilih menu minta rekomendasi. 4. Memasukkan bobot kriteria di form *input* yang telah disediakan. 5. Tekan tombol minta rekomendasi.
Hasil yang Diharapkan	Pengguna dapat membuka form *input* dan mengisi data bobot kriteria di form, kemudian menyimpan data tersebut.

Tabel 6.2 merupakan prosedur kasus uji fungsional memasukkan bobot kriteria. Detail kasus uji memasukkan bobot kriteria terdapat pada Lampiran 2.

Tabel 6.3 Kasus Uji Melihat Hasil Rekomendasi

Nomor Kasus Uji	**KUF-003**
Nama Kasus Uji	**Melihat Hasil Rekomendasi**
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Memastikan bahwa system dapat memenuhi kebutuhan dalam menyediakan hasil rekomendasi tempat wisata pantai.

Prosedur Pengujian	1. Buka aplikasi. 2. Masuk ke halaman *home*. 3. Pilih menu minta rekomendasi. 4. Masuk ke halaman *input* bobot kriteria. 5. Mengisi semua form. 6. Tekan tombol minta rekomendasi.
Hasil yang Diharapkan	Pengguna dapat membuka halaman hasil rekomendasi dan melihat daftar rekomendasi pantai pada halaman tersebut.

Tabel 6.3 merupakan prosedur kasus uji fungsional melihat hasil rekomendasi. Detail kasus uji melihat hasil rekomendasi terdapat pada Lampiran 3.

Tabel 6.4 Kasus Uji Fungsional Menampilkan Informasi Pantai

Nomor Kasus Uji	**KUF-004**
Nama Kasus Uji	**Melihat Informasi Pantai**
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Memastikan bahwa system dapat memenuhi kebutuhan dalam menampilkan informasi terkait pantai secara lebih men-detail.
Prosedur Pengujian	1. Buka aplikasi. 2. Masuk ke halaman *home*. 3. Pilih menu minta rekomendasi. 4. Masuk ke halaman *input* bobot kriteria. 5. Mengisi semua form. 6. Tekan tombol minta rekomendasi. 7. Masuk ke halaman hasil rekomendasi. 8. Memilih salah satu pantai pada halaman hasil rekomendasi.
Hasil yang Diharapkan	Pengguna dapat membuka halaman *detail* pantai dan melihat informasi terkait pantai yang dipilih.

Tabel 6.4 merupakan prosedur kasus uji menampilkan informasi pantai. Detail kasus uji menampilkan informasi pantai terdapat pada Lampiran 4.

Tabel 6.5 Kasus Uji Fungsional Menampilkan Peta

Nomor Kasus Uji	**KUF-005**
Nama Kasus Uji	**Menampilkan Peta**
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Memastikan bahwa system dapat memenuhi kebutuhan dalam menampilkan peta digital *google maps* serta rute menuju lokasi pantai.

Prosedur Pengujian	1. Buka aplikasi. 2. Masuk ke halaman *home*. 3. Pilih menu minta rekomendasi. 4. Masuk ke halaman *input* bobot kriteria. 5. Mengisi semua form. 6. Tekan tombol minta rekomendasi. 7. Masuk ke halaman hasil rekomendasi. 8. Memilih salah satu pantai pada halaman hasil rekomendasi. 9. Tekan tombola rah.
Hasil yang Diharapkan	Pengguna dapat membuka aplikasi *google maps* dan melihat rute menuju lokasi pantai.

Tabel 6.5 merupakan prosedur kasus uji fungsional menampilkan peta. Detail kasus uji menampilkan peta terdapat pada Lampiran 5.

6.2 Pengujian Non-Fungsional

Pada tahap ini terdapat 2 pengujian yang akan dilakukan yaitu validasi algoritma dan *rank consistency*.

6.2.1 Pengujian Validasi Algoritma

Pengujian ini dilakukan dengan membandingkan hasil antara keluaran sistem dan perhitungan manual berupa urutan rekomendasi wisata pantai dari yang teratas sampai terendah dengan data alternatif yang sama. Algoritma dapat dikatakan valid jika hasil dari keluaran system sama dengan hasil dari perhitungan manual.

Tabel 6.6 menunjukkan nilai bobot kriteria yang dipakai pada pengujian. Tabel 6.7 menjelaskan data alternatif yang digunakan untuk pengujian validasi algoritma dan Tabel 6.8 menjelaskan hasil antara keluaran sistem dan perhitungan manual. Hasil perhitungan manual diperoleh dari sub bab 4.2.6.2 perhitungan manual di bab perancangan sebelumnya.

Tabel 6.6 Bobot Kriteria

No	Kriteria	Status	Bobot
1	P	COST	1
2	D	COST	2
3	R	Benefit	2
4	T	Benefit	1
5	F	Benefit	1

Tabel 6.7 Data Alternatif

No	Alternatif Pantai	P	D	R	T	F

1	Pantai Balekambangan	10000	61	4,3	3	6
2	Pantai Ngantep	4000	59	4,3	2	4
3	Pantai Bajulmati	3000	58	4,3	3	3
4	Pantai Goa Cina	4000	72	4,3	3	4
5	Pantai Sendang Biru	6000	74	4,1	3	4

Tabel 6.8 Perbandingan Keluaran Sistem dan Perhitungan Manual

No	Hasil Perhitungan Sistem		Hasil Perhitungan Manual	
	Urutan	Nilai	Urutan	Nilai
1.	Pantai Bajulmati	0.65518522947134	Pantai Bajulmati	0.665393
2.	Pantai Ngantep	0.6518693836659	Pantai Ngantep	0.651692
3.	Pantai Goa Cina	0.6256520030478	Pantai Goa Cina	0.603729
4.	Pantai Sendang Biru	0.48572232218619	Pantai Sendang Biru	0.459254
5.	Pantai Balekambangan	0.42444508252379	Pantai Balekambangan	0.419953

Tabel 6.8 menjelaskan hasil yang diperoleh dari perhitungan metode TOPSIS berupa hasil perhitungan sistem dan hasil perhitungan manual. Data alternatif terdiri dari 5 nama alternatif dan nilai kriteria (P untuk *price,* D untuk *distance,* R untuk *rating,* T untuk transportasi, F untuk fasilitas) tiap alternatif. Hasil perhitungan berupa nama alternatif dan nilai tiap alternatif yang diurutkan dari nilai tertinggi sampai terendah.

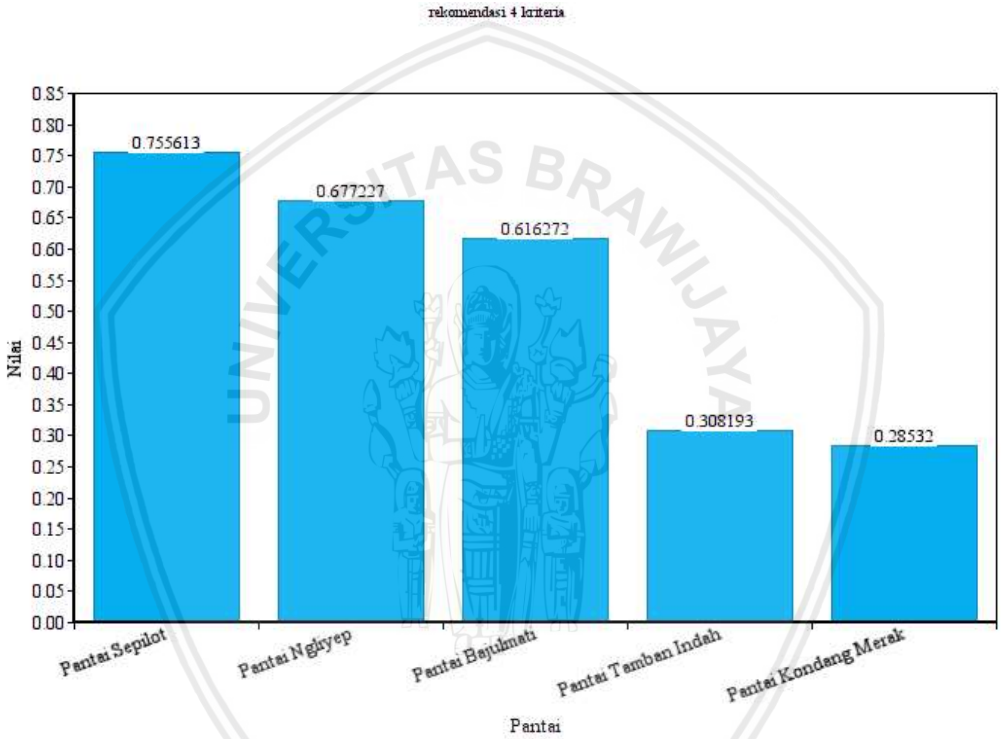
6.2.2 Pengujian *Rank Consistency*

Tujuan dari pengujian ini adalah untuk mengetahui tingkat ketahanan hasil rekomendasi terhadap perubahan yang terjadi. Data awal yang menjadi patokan dalam pengujian kali in merupakan hasil rekomendasi dengan menggunakan 4 kriteria (tanpa kriteria *rating*). Selanjutnya hasil rekomendasi dari 4 kriteria awal akan dibandingkan dengan hasil rekomendasi yang telah dilakukan penamahan dan pengurangan. Pada pengujian *rank consistency* bobot kriteria yang digunakan bernilai 1 (satu) untuk semua kriteria.

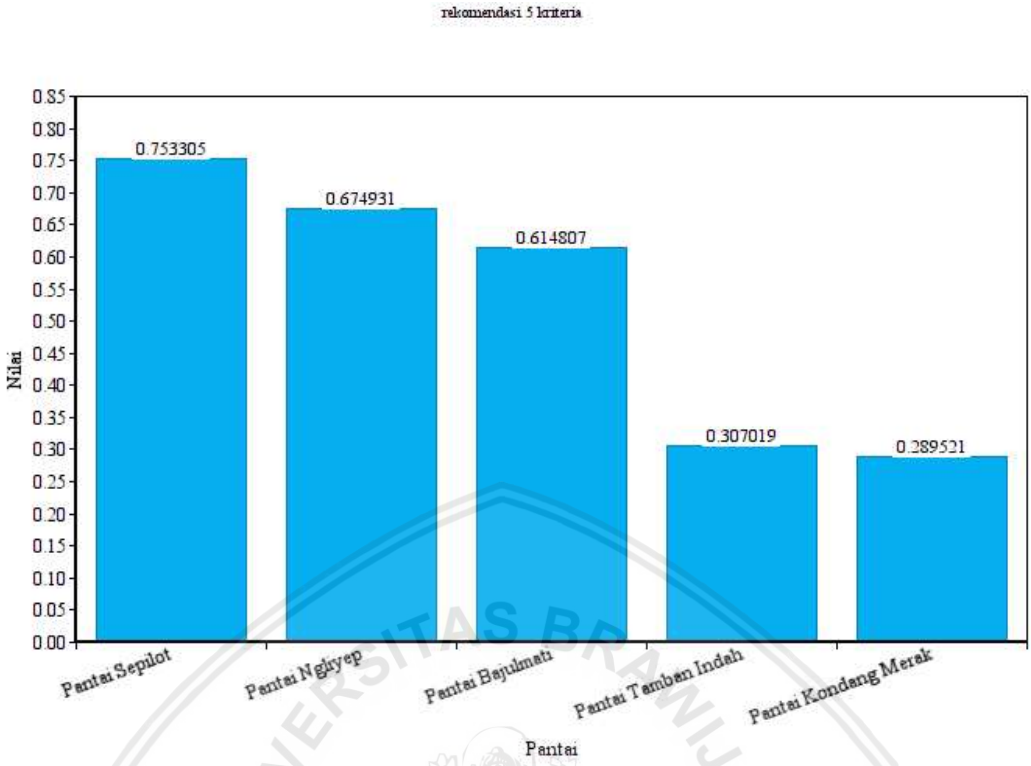
Tabel 6.9 menjelaskan data alternatif yang digunakan untuk pengujian *rank consistency,* Gambar 6.2 menunjukkan hasil rekomendasi 4 kriteria, Gambar 6.3 menunjukkan hasil rekomendasi 5 kriteria (ditambah dengan kriteria *ranking*) dan Gambar 6.4 menunjukkan hasil rekomendasi 3 kriteria (dikurangi kriteria transportasi).

Tabel 6.9 Data Alternatif

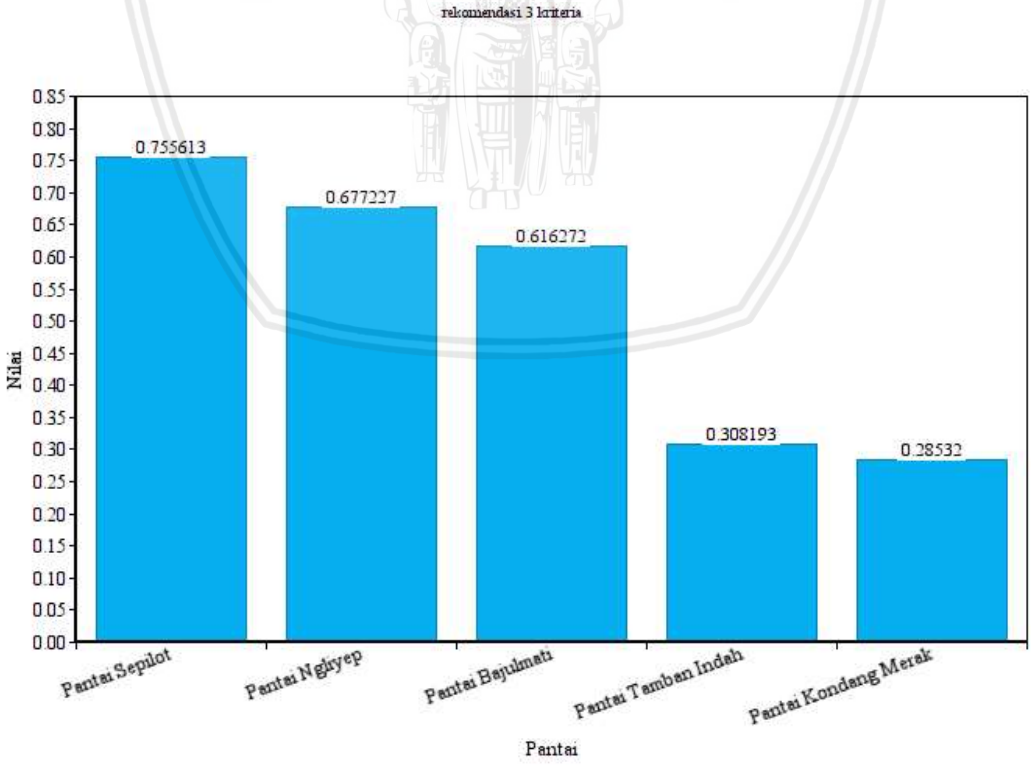
No	Alternatif Pantai	P	D (m)	R	T	F
1	Pantai Ngliyep	3000	67080	4,3	3	4
2	Pantai Kondang Merak	5000	64553	4,5	3	4
3	Pantai Bajulmati	3000	58453	4,3	3	3
4	Pantai Tamban Indah	4000	72429	4,2	3	3
5	Pantai Sepilot	2000	78134	4,3	3	4



Gambar 6.2 Data Hasil Rekomendasi 4 Kriteria



Gambar 6.3 Data Hasil Rekomendasi 5 Kriteria



Gambar 6.4 Data Hasil Rekomendasi 3 Kriteria

Pada saat kriteria ditambah 1, dapat dilihat bahwa hasil rekomendasi tidak terjadi perubahan sama sekali jika dilihat dari urutan rekomendasi. Sedangkan jika dilihat pada nilai yang keluar terdapat perbedaan. Akan tetapi, perbedaan pada nilai yang ada tidak terlalu besar sehingga tidak terlalu berpengaruh pada hasil rekomendasi secara keseluruhan.

Begitupun pada saat kriteria dikurangi 1, dapat dilihat bahwa hasil rekomendasi tidak terjadi perubahan sama sekali jika dilihat dari urutan rekomendasi. Sedangkan jika dilihat pada nilai yang keluar terdapat perbedaan. Akan tetapi, perbedaan pada nilai yang ada tidak terlalu besar sehingga tidak terlalu berpengaruh pada hasil rekomendasi secara keseluruhan.

6.3 Analisis

Mengetahui hasil dan menarik kesimpulan dari hasil pengujian system merupakan tujuan dari proses analisis.

6.3.1 Analisis Pengujian Fungsional

Tabel 6.10 merupakan hasil dari pengujian *blackbox* yang telah dilakukan sebelumnya.

Tabel 6.10 Hasil Pengujian Fungsional

Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status
KUF-001	Pengguna dapat membuka aplikasi dan masuk ke halaman *home*. Pengguna juga dapat melihat daftar pantai pada halaman *home*	Pengguna dapat membuka aplikasi dan masuk ke halaman *home*. Pengguna juga dapat melihat daftar pantai pada halaman *home*	Valid
KUF-002	Pengguna dapat membuka form *input* dan mengisi data bobot kriteria di form, kemudian menyimpan data tersebut.	Pengguna dapat membuka form *input* dan mengisi data bobot kriteria di form, kemudian menyimpan data tersebut.	Valid
KUF-003	Pengguna dapat membuka halaman hasil rekomendasi dan melihat daftar rekomendasi pantai pada halaman tersebut.	Pengguna dapat membuka halaman hasil rekomendasi dan melihat daftar rekomendasi pantai pada halaman tersebut.	Valid

KUF-004	Pengguna dapat membuka halaman *detail* pantai dan melihat informasi terkait pantai yang dipilih.	Pengguna dapat membuka halaman *detail* pantai dan melihat informasi terkait pantai yang dipilih.	Valid
KUF-005	Pengguna dapat membuka aplikasi *google maps* dan melihat rute menuju lokasi pantai.	Pengguna dapat membuka aplikasi *google maps* dan melihat rute menuju lokasi pantai.	Valid

Berdasarkan pada hasil pengujian *blackbox* maka dapat dikatakan bahwa 100% kebutuhan fungsional pada system terpenuhi.

6.3.2 Analisis Pengujian Non-Fungsional

Terdapat 2 bagian analisis pada pengujian non fungsional, yakni: analisis validasi algoritma dan analisis *rank consistency.*

1.3.2.1 Analisis Pengujian Validasi Algoritma

Tabel 6.11 Hasil Pengujian Validasi Algoritma

No	Hasil Perhitungan		Status
	Sistem	Manual	
1.	Pantai Bajulmati	Pantai Bajulmati	Valid
2.	Pantai Ngantep	Pantai Ngantep	Valid
3.	Pantai Goa Cina	Pantai Goa Cina	Valid
4.	Pantai Sendang Biru	Pantai Sendang Biru	Valid
5.	Pantai Balekambangan	Pantai Balekambangan	Valid

Tabel 6.11 menjelaskan hasil dari pengujian validasi algoritma yang telah dilakukan sebelumnya. Berdasarkan hasil pengujian tersebut, maka dapat dikatakan bahwa algoritma yang diterapkan pada system memiliki validitas sebesar 100%.

1.3.2.2 Analisis Pengujian *Rank Consistency*

Tabel 6.12 Hasil Pengujian *Rank Consistency* (Rekomendasi)

No	Hasil Rekomendasi		
	5 Kriteria	4 Kriteria	3 Kriteria
1.	Pantai Sepilot	Pantai Sepilot	Pantai Sepilot
2.	Pantai Ngliyep	Pantai Ngliyep	Pantai Ngliyep

3.	Pantai Bajulmati	Pantai Bajulmati	Pantai Bajulmati
4.	Pantai Tamban Indah	Pantai Tamban Indah	Pantai Tamban Indah
5.	Pantai Kondang Merak	Pantai Kondang Merak	Pantai Kondang Merak

Tabel 6.12 menampilkan hasil pengujian antara metode TOPSIS dengan 5 kriteria, 4 kriteria, dan 3 kriteria. Berdasarkan pengujian ini didapatkan hasil rekomendasi dengan urutan rangking yang sama. Dari hasil pengujian ini, bisa dikatakan bahwa metode TOPSIS merupakan metode yang memiliki tingkat konsistensi yang baik.

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Setelah dilakukan penelitian implementasi TOPSIS pada system rekomendasi tempat wisata pantai di sekitar Malang, maka beberapa kesimpulan yang dapat penulis ambil adalah sebagai berikut:

1. Sistem rekomendasi tempat wisata pantai berbasis lokasi dibangun melalui 5 tahapan, yaitu: studi literatur untuk mempelajari terkait penelitian sebelumnya yang serupa, analisis kebutuhan agar mengetahui sistem yang akan dibuat memiliki fitur-fitur apa saja, perancangan untuk melakukan perancangan berdasarkan analisi kebutuhan yang sudah dibuat, implementasi sistem merupakan tahapan untuk mengimplementasikan sistem sesuai dengan perancangan yang sudah dibuat dan pengujian sistem untuk menguji sistem yang sudah diimplementasikan sebelumnya.
2. Sistem rekomendasi tempat wisata pantai dapat diimplementasikan menggunakan metode TOPSIS dengan 5 parameter kriteria berupa harga, *rating,* jarak, fasilitas dan transportasi. Dimana semua alternative hasil rekomendasi akan ditampilkan dan diurut berdasarkan nilai prefensi tertinggi hingga terendah.
3. Hasil pengujian sistem rekomendasi tempat wisata pantai di sekitar Malang terbagi menjadi tiga, yakni:
    a. Hasil pengujian fungsional dalam sistem rekomendasi tempat wisata pantai di sekitar Malang diperoleh hasil sesuai dengan harapan, dimana semua kebutuhan fungsional terpenuhi.
    b. Pada pengujian validasi algoritma perbandingan antara hasil keluaran system dengan hasil perhitungan manual didapatkan hasil 100% valid.
    c. Hasil pengujian *rank consistency* dari hasil keluaran rekomendasi sistem menunjukkan bahwa metode TOPSIS merupakan metode yang memiliki tingkat konsistensi yang baik.

7.2 Saran

Saran untuk pengembangan system kedepannya, antara lain:

1. Perlu dilakukan pembaruan data pantai secara berkala agar data yang disajikan merupakan data yang terbaru.
2. Perlu digunakan metode atau gabungan metode rekomendasi yang lain untuk meningkatkan hasil rekomendasi.
3. Perlu ditambahkan parameter kriteria agar sistem dapat memahami persepsi keinginan manusia secara lebih baik dan spesifik, sehingga dapat meningkatkan hasil rekomendasi.

DAFTAR PUSTAKA

Arifin, Bustanul. 2015. Sistem Pendukung Keputusan Rekomendasi Pemilhan Mobil Bekas Dengan Menggunakan Metode TOPSIS. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.

BPS, 2018. Statistik Wisatawan 2017-2018. [Online] Available at: <https://www.bps.go.id/dynamictable/2018/04/05/1296/jumlah-kunjungan-wisatawan-mancanegara-per-bulan-ke-indonesia-menurut-pintu-masuk-2017-2018.html>[Diakses 28 Agustus 2018]

Developers, 2018. *Android Developers*. [online] Available at: <https://developer.android.com/> [Accessed 16 Aug. 2018].

Fauzan, Ach. 2017. Pembbangunan Aplikasi Iqro' Berbasis Android Menggunakan Google Speech. Malang: Universitas Brawijaya.

Firmansyah, M.D., Santoso, E. and Dewi, R.K., 2018. Sistem Rekomendasi Tempat Wisata di Kota Malang Menggunakan Metode Hybrid Fuzzy-Floyd Warshall. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(7), pp.2442–2447.

Ginting, G., Fadlina, Mesran, Siahaan, A.P.U. and Rahim, R., 2017. Technical Approach of TOPSIS in Decision Making. *International Journal of Recent Trends in Engineering and Research*, [online] 3(8), pp.58–64. Available at: <http://www.ijrter.com/papers/volume-3/issue-8/technical-approach-of-topsis-in-decision-making.pdf>.

Gunawan. Halim, Fandi. Wilson. 2014. *Penerapan Metode TOPSIS dan AHP pada Sistem Penunjang Keputusan Penerimaan Anggota Baru, Studi Kasus: Ikatan Mahasiswa Sistem Informasi STMIK Mikroskil Medan*. Medan: STMIK Mikroskil.

Larasati, A.A., Setianingrum, A.H. and Wardhani, L.K., 2017. Development decision support system of choosing medicine using TOPSIS method (Case study: RSIA Tiara). *Proceedings - 6th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2016*, pp.160–165.

Lokare, V.T. and Jadhav, P.M., 2016. Using the AHP and TOPSIS methods for decision making in best course selection after HSC. In: *2016 International Conference on Computer Communication and Informatics, ICCCI 2016*.

Maharani, Septya. Hatta, Heliza R. Merdiko, Gery. 2014. *Decision Support System Of Culinary Recomendations Using AHP and TOPSIS Methods With Map Visualization*. FMIPA, Universitas Mulawarman

Noor, A.S., 2012. Pengembangan Pemanfaatan Peta Digital Untuk Mendukung Sistem Informasi Pari- Di Kabupaten Malang. pp.1–11.

Perdana, Reza. 2014. *Sistem Rekomendasi Tempat Wisata Kuliner Menggunakan Metode TOPSIS Berbasis Mobile.* Malang: Universitas Islam Negeri Maulana Malik Ibrahim.

Ramadhan, Dayan Ramly. 2015. Pengembangan dan Analisis Kualitas Aplikasi Mobile School Maps (*MooMaps)* Berbasis Mobile Application untuk Pemetaan Universitas di Yogyakarta. Yogyakarta: Universitas Negeri Yogyakarta.

Risza, Nurrachman. 2016. Implementasi Metode AHP-TOPSIS Pada Sistem Rekomendasi Kuliner. Malang: Universitas Brawijaya.

Rodrigues, F., Junior, L., Osiro, L., Cesar, L. and Carpinetti, R., 2014. A comparison between Fuzzy AHP and Fuzzy TOPSIS methods to supplier selection. *Applied Soft Computing Journal*, [online] 21, pp.194–209. Available at: <http://dx.doi.org/10.1016/j.asoc.2014.03.014>.

Safaat, Nazruddin. 2015. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Andoid.* Bandung: Informatika.