



IMPLEMENTASI SISTEM REKOMENDASI TEMPAT PEMBELIAN OLEH-OLEH KHAS MALANG BERBASIS PERANGKAT BERGERAK

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Heru Putra

NIM: 155150200111304



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020

PENGESAHAN

**IMPLEMENTASI SISTEM REKOMENDASI TEMPAT PEMBELIAN OLEH-OLEH KHAS
MALANG BERBASIS PERANGKAT BERGERAK**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

Disusun Oleh :

Heru Putra

NIM: 155150200111304

**Skripsi ini telah diuji dan dinyatakan lulus pada
11 Juni 2020**

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

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

Komang Candra Brata, S.Kom., M.T., M.Sc.
NIK: 201607 890711 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Wahoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

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

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

Malang, 12 Juni 2020



Heru Putra

NIM: 155150200111304

ABSTRAK

Heru Putra, Implementasi Sistem Rekomendasi Tempat Pembelian Oleh-Oleh Khas Malang Berbasis Perangkat Bergerak

Pembimbing: Ratih Kartika Dewi, S.T., M.Kom. Dan Komang Candra Brata, S.Kom., M.T., M.Sc.

Buah tangan atau dikenal dengan oleh oleh biasa diberikan oleh orang yang baru pulang dari bepergian. Pemberian buah tangan ini biasa diberikan untuk mempererat pertemanan dan persaudaraan. Buah tangan atau oleh oleh itu bisa berupa makanan khas daerah tertentu atau makanan yg sedang digemari saat ini. Namun, perlu kehati-hatian untuk menentukan jenis oleh oleh yang akan dibeli, baik dari sisi harga, kualitas, dan kadaluarsanya. Dengan adanya permasalahan tersebut, pengembangan dari sistem rekomendasi oleh-oleh dikemukakan untuk mempermudah dalam mencari sebuah oleh-oleh yang tepat. Pengembangan Sistem Rekomendasi Oleh-oleh ini menggunakan salah satu *device mobile* yang berbasis Android OS. Sistem rekomendasi oleh-oleh ini menggunakan metode TOPSIS untuk memberikan hasil rekomendasi dari inputan pengguna. Pengembangan ini juga menggunakan fitur seperti sensor GPS untuk melakukan pemetaan posisi pengguna sampai tujuan dari rekomendasi yang akan ditentukan. Pengujian fungsional dari sistem rekomendasi oleh-oleh yang dibangun ini telah memenuhi kevalid-an dengan nilai yaitu 100%. Begitu juga dengan pengujian usability menggunakan *System Usability Scale* dengan nilai hasil yaitu 75,5 dimana nilai tersebut termasuk dalam range *acceptable*.

Kata Kunci: TOPSIS, Android, Oleh-oleh



ABSTRACT

Heru Putra, Implementasi Sistem Rekomendasi Tempat Pembelian Oleh-Oleh Khas Malang Berbasis Perangkat Bergerak

Supervisors: Ratih Kartika Dewi, S.T., M.Kom. Dan Komang Candra Brata, S.Kom., M.T., M.Sc.

Souvenir or we might familiar with "oleh-oleh" is a gift from someone who travel to some place. Souvenir usually being given to other people for increase their relationship to someone that being said they knew or someone we care. There's many type of souvenir that can be given to, such as food that only produce locally or food that is popular among others. However, choosing souvenir, like food, needs a careful approach for instance what kind of souvenir that we wanted to bought, how much it's cost, and how long shoult it takes before the expiration date occurs. As the problem like that rise, recomendation system is required for searching a souvenir that we wanted for people that we care about. The development of Recommendation System for food based Souvenir is for mobile device which using Androd Operating System. This Recomendation System used a method called TOPSIS for the recomendation of food based souvenir. GPS being used in this development for seacrhing a position of the user to reach their desire place from the result of recommendation system. After the development have been implement, functional testing is required to check how this aplication run as it is or not which is the test that already occurred reach a conclusion at 100% valid. For the usability system using System Usatbility Scale testing, the score reach 75.5 which is in a range of acceptable

Keywords: TOPSIS, Android, Oleh-oleh

**DAFTAR ISI**

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR KODE	xiv
DAFTAR LAMPIRAN	xv
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	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Sistem Pendukung Keputusan	5
2.2.1 Definisi Keputusan	6
2.2.2 Teori Pengambilan Keputusan	6
2.2.3 Tujuan Sistem Pendukung Keputusan	6
2.2.4 Tahapan Sistem Pendukung Keputusan	6
2.3 Metode Technique For Order Preference by Similarity to Ideal Solution (TOPSIS)	7
2.3.1 Definisi TOPSIS	7
2.3.2 Langkah-Langkah TOPSIS	7
2.4 Pengujian Perangkat Lunak	9
2.4.1 Pengujian <i>Black-Box</i>	9



2.4.2 System Usability Scale.....	10
2.5 Global Positioning System (GPS).....	11
2.6 Android.....	11
2.7 Javascript Object Notation (JSON).....	12
2.8 Waterfall.....	13
BAB 3 METODOLOGI PENELITIAN.....	15
3.1 Metode Penelitian.....	15
3.2 Tipe Penelitian.....	16
3.3 Subjek, Lokasi dan Data Penelitian.....	16
3.4 Metode Pengumpulan Data.....	16
3.5 Peralatan Pendukung.....	17
3.6 Metode Pengembangan Sistem.....	17
BAB 4 ANALISIS DAN PERANCANGAN.....	18
4.1 Analisis Kebutuhan.....	19
4.1.1 Gambaran Aplikasi.....	19
4.1.2 Identifikasi Aktor.....	20
4.1.3 Analisis Kebutuhan Fungsional.....	20
4.1.4 Analisis Kebutuhan Non-Fungsional.....	24
4.2 Perancangan Perangkat Lunak.....	24
4.2.1 Perancangan Algoritme.....	24
4.2.2 Perancangan Activity Diagram.....	32
4.2.3 Perancangan Arsitektur Diagram.....	34
4.2.4 Perancangan Sequence Diagram.....	35
4.2.5 Perancangan Class Diagram.....	37
4.2.6 Perancangan Basis Data.....	38
4.2.7 Perancangan User Interface.....	39
BAB 5 IMPLEMENTASI.....	43
5.1 Spesifikasi Sistem.....	43
5.1.1 Spesifikasi Perangkat Keras.....	43
5.1.2 Spesifikasi Perangkat Lunak.....	44
5.2 Implementasi Basis Data.....	45
5.3 Implementasi Class dan Assets pada File Program.....	45



5.4 Implementasi Kode Program	46
5.4.1 Implementasi Kode Program Aplikasi Perangkat Bergerak	46
5.4.2 Implementasi Kode Program Algoritme TOPSIS	49
5.5 Implementasi Antarmuka	53
BAB 6 PENGUJIAN DAN ANALISIS	57
6.1 Pengujian <i>Black-Box</i>	57
6.2 Pengujian <i>System Usability Scale</i>	63
6.3 Pengujian Validasi Algoritme	64
BAB 7 KESIMPULAN DAN SARAN	67
7.1 Kesimpulan	67
7.2 Saran	67
DAFTAR REFERENSI	68
LAMPIRAN A PENGUJIAN SYSTEM USABILITY SCALE	71
LAMPIRAN B KUESIONER KEBUTUHAN	76
LAMPIRAN C HASIL KELUARAN ALGORITME	79



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka.....	5
Tabel 4.1 Identifikasi Aktor.....	20
Tabel 4.2 Kebutuhan Fungsional.....	20
Tabel 4.3 <i>Use Case Scenario</i> Memasukkan Bobot Kriteria Oleh-oleh.....	22
Tabel 4.4 <i>Use Case Scenario</i> Melihat Rekomendasi Tempat Oleh-oleh.....	22
Tabel 4.5 <i>Use Case Scenario</i> Melihat Detail Tempat Oleh-oleh.....	23
Tabel 4.6 Skenario <i>Use Case</i> Melihat Lokasi Tempat Oleh-oleh.....	23
Tabel 4.7 Skenario <i>Use Case</i> Mengubah Detail Tempat Oleh-oleh.....	23
Tabel 4.8 Spesifikasi Kebutuhan Non Fungsional.....	24
Tabel 4.9 Data Lokasi.....	27
Tabel 4.10 Data <i>Sample</i>	27
Tabel 4.11 Hasil Penjumlahan Alternatif Tiap Kriteria.....	28
Tabel 4.12 Hasil Normalisasi Matriks Alternatif.....	28
Tabel 4.13 Hasil Normalisasi Matriks Terbobot.....	29
Tabel 4.14 Hasil Solusi Ideal Positif dan Solusi Ideal Negatif.....	29
Tabel 4.15 Hasil Perhitungan Jarak Terhadap Solusi Ideal Positif dan Negatif.....	30
Tabel 4.16 Hasil Perhitungan Nilai Preferensi.....	31
Tabel 4.17 Hasil Perangkingan.....	31
Tabel 4.18 Perancangan Basis Data tempat_oleh_oleh.....	38
Tabel 4.19 Perancangan Basis Data admin_oleh_oleh.....	38
Tabel 5.1 Spesifikasi Perangkat Keras <i>Laptop</i>	44
Tabel 5.2 Spesifikasi <i>Mobile Device</i>	44
Tabel 5.3 Spesifikasi Perangkat Lunak pada <i>Laptop</i>	44
Tabel 5.4 Spesifikasi Perangkat Lunak pada <i>Mobile Device</i>	44
Tabel 5.5 Berkas Kelas dan <i>Assets Layout</i> pada Aplikasi Perangkat Bergerak.....	46
Tabel 6.1 Pengujian Fungsional Memasukkan Bobot Kriteria.....	57
Tabel 6.2 Pengujian Fungsional Melihat Hasil TOPSIS.....	58
Tabel 6.3 Pengujian Fungsional Melihat Detail Rekomendasi.....	59
Tabel 6.4 Pengujian Fungsional Melihat Lokasi Hasil Rekomendasi.....	60
Tabel 6.5 Pengujian Fungsional Melihat Lokasi Hasil Rekomendasi.....	62



Tabel 6.6 Pengujian SUS.....	63
Tabel 6.6 Bobot Kriteria Pengujian.....	64
Tabel 6.7 Alternatif Data Pengujian.....	64
Tabel 6.8 Perbandingan Hasil Antara Dua Perhitugan.....	65



DAFTAR GAMBAR

Gambar 2.1 System Usability Scale.....	11
Gambar 2.2 Struktur Objek Dalam Format JSON.....	12
Gambar 2.3 Struktur Array Dalam Format JSON.....	13
Gambar 2.4 Struktur Nilai Dalam Format JSON.....	13
Gambar 2.5 Metode Waterfall.....	14
Gambar 3.1 Alur Metodologi Penelitian.....	15
Gambar 4.2 Gambaran Aplikasi Rekomendasi Oleh-oleh.....	19
Gambar 4.3 Use Case Diagram Kebutuhan Fungsional.....	21
Gambar 4.4 Flow Chart Haversine.....	25
Gambar 4.5 Flow Chart TOPSIS.....	26
Gambar 4.6 Activity Diagram Memasukkan Bobot.....	32
Gambar 4.7 Activity Diagram Menampilkan Hasil Rekomendasi.....	33
Gambar 4.8 Activity Diagram Mengubah Detail Tempat Oleh-Oleh.....	33
Gambar 4.9 Arsitektur Perangkat Bergerak.....	34
Gambar 4.10 Sequence Diagram Memasukkan Bobot dan Hasil Rekomendasi.....	35
Gambar 4.11 Sequence Diagram Melihat Ringkasan.....	36
Gambar 4.12 Sequence Diagram Mengubah Detail Tempat Oleh-oleh.....	36
Gambar 4.13 Class Diagram Perangkat Bergerak Sistem Rekomendasi Tempat Oleh-oleh.....	37
Gambar 4.14 Halaman Memasukkan Bobot dan Melihat Rekomendas.....	39
Gambar 4.15 Halaman Melihat Rekomendasi dan Detail Tempat.....	40
Gambar 4.16 Halaman Melihat Detail tempat dan melihat lokasi tempat.....	41
Gambar 4.17 Halaman login dan Mengubah detail Oleh-oleh.....	41
Gambar 5.1 Diagram Pohon Implementasi.....	43
Gambar 5.2 Halaman Memasukkan Bobot.....	53
Gambar 5.3 Halaman Melihat Rekomendasi.....	54
Gambar 5.4 Halaman Melihat Detali tempat.....	55
Gambar 5.5 Halaman Melihat Detali tempat.....	55
Gambar 5.6 Halaman Login.....	56
Gambar 5.5 Halaman Mengubah Detail Tempat Oleh-oleh.....	56



Gambar 6.1 Pohon Diagram Pengujian dan Analisis.....	57
Gambar 6.2 Hasil Pengujian Masukkan bobot.....	58
Gambar 6.3 Hasil Pengujian Melihat hasil TOPSIS.....	59
Gambar 6.4 Hasil Pengujian Melihat detail hasil rekomendasi.....	60
Gambar 6.5 Hasil Pengujian Melihat lokasi tempat rekomendasi.....	61
Gambar 6.6 Hasil Pengujian Mengubah Detail Tempat oleh-oleh.....	62



DAFTAR KODE

Kode 5.1 Perintah SQL Membuat Tabel <i>topsis_oleh_oleh</i>	45
Kode 5.2 Perintah SQL Membuat Tabel <i>admin_oleh_oleh</i>	45
Kode 5.3 Kode Program Kelas <i>MainActivity</i>	47
Kode 5.4 Kode Program Kelas <i>HasilTopsisActivity</i>	48
Kode 5.5 Kode Program Kelas <i>DetailHasilActivity</i>	49
Kode 5.6 Kode Program Kelas <i>AdminActicity</i>	49
Kode 5.7 Kode Program Menghitung nilai Haversine.....	50
Kode 5.8 Kode Program Menghitung nilai pembagi.....	50
Kode 5.9 Kode Program Menghitung Nilai Normalisasi.....	50
Kode 5.10 Kode Program Menghitung Nilai Pembobot.....	51
Kode 5.11 Kode Program Menghitung Nilai a+ dan a- dengan si+ dan si.....	51
Kode 5.12 Kode Program Menghitung jumlah SI dan nilai ci+.....	52
Kode 5.13 Kode Program Mengurutkan Nilai dari ci+.....	52

**DAFTAR LAMPIRAN**

LAMPIRAN A PENGUJIAN <i>SYSTEM USABILITY SCALE</i>	71
A.1 Responden Satu	71
A.2 Responden Dua	72
A.3 Responden Tiga	73
A.4 Responden Empat	74
A.5 Responden Lima	75
LAMPIRAN B KUESIONER KEBUTUHAN	76
B.1 Gambar Satu	76
B.2 Gambar Dua	76
B.3 Gambar Tiga	77
B.4 Gambar Empat	77
B.5 Gambar Lima	78
LAMPIRAN C HASIL KELUARAN ALGORITME	79
C.1 JSON Hasil Penghitungan Algoritme	79



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Oleh – oleh, barang maupun makanan ringan yang biasanya dibawa oleh orang yang sedang berpergian (KBBJ, 2016). Terkadang, oleh-oleh disebut sebagai cinderamata, yaitu sebagai sebuah pemberian dimana oleh-oleh adalah sesuatu yang dibagikan kepada orang sebagai kenang-kenangan dari orang yang berpergian.

Oleh-oleh digemari oleh banyak kalangan masyarakat Indonesia. Setiap kali ada kerabat maupun saudara yang dikenal akan berpergian, disanalah orang akan meminta oleh-oleh. Memang permintaan tersebut sudah tidak jarang di kalangan masyarakat dan bahkan membuat orang yang berpergian harus membawa barang yang begitu banyak saat hendak pulang ke daerah asalnya.

Ada 14 atribut atau faktor dalam pembelian oleh-oleh yaitu harga, rasa, aroma, merek, kemasan, jaminan halal, volume bersih, informasi nutrisi, komposisi, masa kadaluarsa, higienis, promosi, ketersediaan produk dan lokasi penjualan (Purba, 2016). Untuk batasan faktor, mulai dari harga maupun tanggal kadaluarsa untuk barang seperti makanan ringan dan lokasi tempat penjualan hanya digunakan dikarenakan kurangnya informasi secara *online* pada masing-masing tempat pembelian tersebut.

Harga itu sendiri menjadi indikator dikarenakan bila konsumen harus membayar lebih dari manfaat yang diterimanya, maka konsumsi pada produk tersebut akan berkurang (Ghanimata, 2012). Kadaluarsa juga menjadi hal yang diperhatikan dan menjadi salah satu atribut dalam pengambilan keputusan pada sebuah produk makanan (Purba, 2016). Kita juga tidak ingin memberikan kekerabat kita barang yang kurang menarik maupun makanan yang kadaluarsa.

Kurangnya informasi tentang dimana saja tempat oleh-oleh maupun apa saja barang yang akan dibeli merupakan kelemahan dari pada wisatawan maupun masyarakat sekitar. Dengan begitu, pembuatan aplikasi sistem rekomendasi tempat oleh-oleh khas Malang ini bertujuan untuk mempermudah wisatawan maupun masyarakat Malang. Dengan bantuan teknologi GPS zaman sekarang, wisatawan maupun masyarakat dapat dengan mudah mencari tempat oleh-oleh sesuai dengan keinginan seperti dekatnya tempat oleh-oleh dari orang tersebut, dimana dengan jarak tempuh yang lebih dekat akan menjadi lebih hemat dalam penggunaan biaya, waktu maupun tenaga (Purnawan, 2018).

Survei dari 40 responden yang terdiri dari orang yang berumur 17 – 25 tahun, bahwa sebanyak 75% dari mereka sering membelikan oleh-oleh. Beberapa mengatakan bahwa masih banyak orang bimbang dalam mencari sebuah oleh-oleh. Dari mulai harga yang belum diketahui, kadaluarsa dari sebuah produk yang belum diketahui dan tempat membeli oleh-oleh yang terjangkau.

Sudah banyak pengembangan untuk membuat sebuah rekomendasi atas sebuah sistem. Pada pengembangan “Sistem Pendukung Keputusan Rekomendasi



Makanan Khas Sulawesi Utara yang menunjang Diet” (Sawel, 2016), studi kasus pada penelitian tersebut adalah memberikans daftar makanan untuk user yang sedang melakukan diet. Dengan kriterianya yaitu protein, karbohidrat dan lemak dimana tiap user memiliki bobot kriteria yang berbeda beda tergantung dari tinggi, berat badan dan jenis kelamin. Pengembangan Sitem rekomendasi tersebut menggunakan metode *AHP* dengan 14 data uji yang ada. Hasil dari pengembangan tersebut adalah daftar makanan dari 14 data uji tersebut dan diberikan ranking tiap makanannya dengan skor paling tinggi yaitu 54.49 dengan nama data uji Nasi Jaha. Dengan skor tersbut, Nasi Jaha disimpulkan menjadi menu makanan yang baik untuk user yang sedang diet.

Pengembangan kedua dengan judul “Implementasi TOPSIS Pada Sistem Rekomendasi Tempat Wisata Pantai Di Sekitar Malang Berbasis Lokasi” (memiliki studi kasus pemberian rekomendasi tempat wisata pantai sekitar Malang dengan kriteria *price, distance, rating, transportasi* dan fasilitas. Pada pengembangan ini, metode yang digunakan adalah TOPSIS. Hasil dari pengembangan ini yaitu menampilkan data tempat rekomendasi wisata pantai sekitar malang dengan acuan bobot yang diisikan pada kriteria tiap usernya.

Dari metode-metode yang sudah ada, metode untuk pengembangan Rekomendasi oleh oleh camilan khas malang adalah TOPSIS. *Technique for Order of Preference by Similarity to Ideal Solition* adalah salah satu metode untuk melakukan pemilihan putusan dari banyak kriteria atau dapat disebut dengan *multi-criteria decision analyst* (MCDA). TOPSIS juga memiliki tingkat kompleksitas yang rendah, sehingga dapat diterapkan pada perangkat bergerak. Selain tingkat kompleksitas yang rendah, TOPSIS memiliki *error rate* yang rendah dibandingkan dengan metode WP (*Weighted Product*) dengan *error rate* dari TOPSIS yaitu 27% dan *error rate* dari WP yaitu 33% (Purwandi, 2019). Metode ini juga meliki performa yang lebih unggul daripada metode lain, seperti AHP dan juga fuzzy (Dewi, 2018).

Berdasarkan permasalahan diatas, diketahui bahwa kurangnya informasi tentang letak, jenis, maupun harga oleh-oleh. Dengan begitu peneliti ingin mengembangkan aplikasi tentang sistem rekomendasi tempat oleh-oleh khas malang berbasis android menggunakan metode TOPSIS karena peforma yang mencukupi pada perangkat berbasis bergerak. Pada perangkat bergerak ini, tedapat admin sebagai pengguna yang melakukan perubahan sesuai dengan tempat oleh-oleh tersebut untuk mempermudah pencarian secara *real-time*.

Pengujian pengukuran aplikasi sistem rekomendasi oleh-oleh ini menggunakan *black-box* dimana pengujian tersebut digunakan untuk validasi dari sistem yang dibuat apakah sistem tersebut sudah dapat berjalan dengan baik. Dengan begitu pengujian ini digunakan sebagai tolak ukur apakah sistem yang dikerjakan ini sudah sesuai dengan kebutuhan yang diberikan. Pengujian *black-box* ini merupakan metode pengujian untuk fungsionalitas dari kebutuhan yang telah ditentukan.



Dilaksanakan juga pengujian *System Usability Scale* (SUS) untuk dilakukan sebuah pengukuran apakah aplikasi pada perangkat bergerak ini dapat diterima oleh pengguna (Ependi, 2017). Dengan begitu, pengujian SUS ini diperulka untuk mengetahui bagaimana kelayakan dari sistem ini dengan dapatnya diterima atau tidak dengan jumlah reseponden yaitu lima personel. SUS ini digunakan untuk menguji *usability* dari sistem yang dikembangkan.

1.2 Rumusan Masalah

Terdapat masalah-masalah yang dapat dirumuskan pada penelitian. Rumusan masalah tersebut yaitu:

1. Bagaimana kebutuhan aplikasi sistem rekomendasi tempat oleh-oleh khas daerah malang dapat mempermudah pencarian oleh-oleh sesuai dengan keinginan pengguna?
2. Bagaimana merancang aplikasi sistem rekomendasi tempat oleh-oleh khas malang untuk menyelesaikan masalah pencarian oleh-oleh?
3. Bagaimana metode TOPSIS dapat menjadi algoritme untuk pencarian tempat oleh-oleh khas Malang?
4. Bagaimanakah sistem rekomendasi tempat untuk oleh-oleh khas malang diimplementasikan sesuai dengan kebutuhan pengguna?
5. Bagaimana hasil pengujian validasi algoritma, *blackbox*, *System Usability Scale* dari sistem rekomendasi tempat untuk oleh-oleh khas daerah malang setelah melakukan implementasi?

1.3 Tujuan

Tujuan untuk dicapai pada penelitian ini yaitu sebagai berikut:

1. Berhasil Mengimplementasikan sistem rekomendasi oleh oleh khas malang pada perangkat android dimana menampilkan hasil sesuai dengan input pengguna
2. Berhasil Menerapkan metode TOPSIS untuk memberikan rekomendasi oleh-oleh sesuai dengan kebutuhan pengguna
3. Berhasil mendapatkan hasil uji yang dapat menyesuaikan dengan kebutuhan pengguna saat mencari sebuah rekomendasi tempat oleh-oleh.

1.4 Manfaat

Dari penelitian yang dikembangkan ini, diharapkan masyarakat maupun wisatawan bisa didapatkan manfaatnya. Manfaatnya tersebut yaitu:

1. Mempermudah pencarian oleh-oleh sesuai dengan menggunakan perangkat bergerak berbasis Android.
2. Memberikan informasi tentang rekomendasi tempat oleh-oleh pada Google Maps, detail tempat oleh-oleh dan daftar oleh-oleh yang disediakan.



1.5 Batasan Masalah

Batasan masalah dikemukakan untuk melakukan pembatasan dari penelitian agar tidak melebar. Batasan masalah tersebut:

1. Data dari objek oleh-oleh hanya di sekitar Malang.
2. Data dari objek oleh-oleh hanya berjenis camilan
3. Kriteria untuk rekomendasi oleh-oleh yang digunakan adalah harga, lama kadaluarsa, dan jarak ke lokasi oleh-oleh.
4. Metode untuk pengerjaan pengembangan system rekomendasi adalah TOPSIS.
5. Hanya bisa digunakan pada perangkat bergerak berbasis android

1.6 Sistematika Pembahasan

Dalam sistematika penulisan, tersedia gambaran yang dibahas dalam penulisan skripsi. Diantaranya yaitu:

BAB I PENDAHULUAN

Berisi penulisan rumusan penelitian, batasan permasalahan, tujuan meneliti pengembangan ini, manfaat dari penelitian yang dikembangkan, dan sistematika dari Implementasi Sistem Rekomendasi ini.

BAB II TINJAUAN PUSTAKA

Dijelaskan kajian apa saja yang ada berupa tiga macam penelitian yang berkaitan dengan penelitian ini dan dasar teori untuk meneliti.

BAB III METODOLOGI PENELITIAN

Pada bab ini dijelaskan metode apa saja dan bagaimana langkah untuk pengerjaan penelitian ini.

BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini terdapat dua bagian dimana bagian tersebut adalah analisis kebutuhan dan perancangan perangkat lunak.

BAB V IMPLEMENTASI

Pada bab ini, dibahas tentang pengimplementasian perangkat lunak yang sudah dirancang sebelumnya.

BAB VI PENGUJIAN

Dijelaskan bagaimana aplikasi dapat berjalan dengan sesuai menggunakan pengujian blackbox yaitu pengujian fungsionalitas dari aplikasi perangkat lunak tersebut lalu pengujian SUS dan pengujian validasi algoritme agar pengeluaran dari algoritme TOPSIS sesuai dengan hasil yang diinginkan.

BAB VII PENUTUP

Penutup dari penulisan skripsi ini dimana terdapat kesimpulan yang dikembangkan dan saran.



BAB 2 LANDASAN KEPUSTAKAAN

Penjelasan pada bab ini mengacu terhadap hal apa saja sebagaimana dapat membantu pengerjaan laporan dari penelitian rekomendasi ini. Kajian pustaka tersebut membahas tentang penelitian yang sudah pernah dikembangkan dan dijadikan acuan penelitian ini. Teori yang dibutuhkan untuk merancang aplikasi atau perangkat lunak maupun pelaporan ini dibahas pada dasar teori.

2.1 Kajian Pustaka

Tabel 2.1 Kajian Pustaka

No	Peneliti	Judul	Hasil
1	Fernando D. Sawel, Alicia A. E. Sinsuw, Muhamad D. Putro (Sawel, 2016)	Sistem Pendukung Keputusan Rekomendasi Makanan Khas Sulawesi Utara yang Menunjang Diet	Sistem yang dapat memberikan rekomendasi makanan khas Sulawesi utara yang dapat menunjang diet.
2	Muhamad Hilmi Habitullah, Ratih Kartika Dewi, Marji Marji (Habitullah, 2018)	Implementasi Topsis Pada Sistem Rekomendasi Tempat Wisata Pantai Di Sekitar Malang Berbasis Lokasi	Sitem rekomendasi Wisata Malang dengan basis Pantai
3	Dewi, R.K., Hanggara, B.T., Pinandito, A. (Dewi, 2018)	<i>A Comparison Between AHP and Hybrid AHP for Mobile Based Culinary Recommendation System</i>	Hasil Performansi AHP dan Hybrid AHP untuk sistem pencarian makanan berbasis <i>Mobile</i> .

2.2 Sistem Pendukung Keputusan

Decision Support System (DSS) biasa disebut Sistem pendukung keputusan merupakan sistem berbasis komputer yang bersifat interaktif dimana dapat membantu menyelesaikan masalah bersifat kurang terstruktur menggunakan data dan model yang tersedia (Subakti, 2002). Istilah Management Decision System merupakan konsep pertama kali yang dikemukakan pada tahun 1970-an oleh Michael S.Scott Morton. Konsep Management Decision System menurut Michael



adalah suatu sistem berbasis komputer dengan memanfaatkan data-data dan model dimana digunakan sebagai piranti sebuah pengambilan keputusan terkait dengan permasalahan yang tidak terstruktur secara interaktif (Magdalena, 2012). Beberapa ahli juga melakukan pendefinisian pendukung keputusan dimana menurut Herman Rizani, sistem di organisasi untuk mengambil keputusan (Rohayani, 2013).

2.2.1 Definisi Keputusan

Keputusan diambil dari beberapa alternatif pada situasi yang tidak pasti. Pengelolaan situasi yang tidak pasti tersebut harus diambil dengan sebuah keputusan yang menarik perhatian. Keputusan itu sendiri menurut KBBI sangat berhubungan dari putusan yang sudah dipertimbangkan, diperhitungkan dan dipikirkan sebelumnya. Pada buku yang dikemukakan oleh Simon pada tahun 1977 dituliskan bahwa jenis dari keputusan dibagi menjadi dua jenis, yaitu keputusan terprogram dimana keputusan tersebut sudah berjalan berulang kali dan keputusan tidak terprogram yaitu keputusan yang belum pernah sebelumnya diketahui dan bersifat tidak terstruktur (Wulandari, 2020).

2.2.2 Teori Pengambilan Keputusan

Untuk *decision making* memiliki teori-teori yang dikemukakan. Teori pengambilan keputusan adalah sebuah kegiatan pengambilan keputusan dengan panduan tertentu kepada orang-orang atau organisasi yang sedang mengatasi sebuah situasi. Ada tiga kondisi dari pengambilan keputusan, kondisi tersebut yaitu: kondisi pasti, kondisi tidak pasti dan kondisi beresiko. Kondisi pertama yaitu kondisi pasti, dimana setiap faktor yang mempengaruhi pengambilan dari keputusan tersebut memiliki informasi yang lengkap dari situasi yang ada. Untuk kondisi yang kedua yaitu kondisi tidak pasti adalah pengambilan keputusan dimana faktor yang ditimbang tidak diketahui secara jelas. Kondisi yang terakhir adalah saat berisiko dimana tiap alternatif memiliki risiko yang berbeda-beda dari tiap keadaan (Nurrachman, 2016).

2.2.3 Tujuan Sistem Pendukung Keputusan

Menurut Efraim Turban dan Jay E Aronson dalam penjelasan bukunya, sistem pendukung keputusan memiliki tujuan tertentu. Tujuan dari sistem pendukung tersebut: (Rohayani, 2013)

1. Memberikan bantuan dalam pengambilan sebuah keputusan.
2. Meningkatkan efektivitas pengambilan keputusan yang oleh pimpinan.
3. Dengan SPK (Sistem Pendukung Keputusan) yang bersifat digital/computer dapat biaya untuk setiap anggota kelompok. Dimana akan meningkatkan produktivitas pada suatu organisasi.

2.2.4 Tahapan Sistem Pendukung Keputusan

Tahapan dari prosedur sistem pendukung keputusan tersebut terdiri dari empat tahapan yaitu: kecerdasan, desain, pilihan, dan implementasi. Tahapan tersebut dijelaskan sebagai berikut: (Magdalena, 2012)



1. *Intelligent* (Kecerdasan).

Terjadi perlakuan pengenalan, *searching and surfing* untuk proses masalah maupun lingkup masalah.

2. *Design* (Desain)

Terjadi proses untuk melakukan pengembangan dan penemuan alternatif.

3. *Choice* (Pilihan)

Terjadi pemilihan dari alternatif yang sesuai untuk dijalankan dimana meliputi evaluasi, pencarian mapun rekomendasi dari solusi yang terbaik untuk model yang ada.

4. *Implementation* (Implementasi)

Terjadi pelaksanaan keputusan melalui tahap satu sampai tiga.

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

Hwang dan Yoon mengemukakan TOPSIS pada tahun 1981 dimana gagasan utama dari TOPSIS tersebut berasal dari konsep kompromi alternatif dimana terdapat solusi ideal positif dan solusi ideal negatif (Gunawan, 2014).

2.3.1 Definisi TOPSIS

Ada berbagai metode pengambilan sebuah keputusan, salah satunya adalah TOPSIS. TOPSIS itu sendiri adalah metode pengambilan keputusan dimana sebuah masalah tersebut memiliki multikriteria dengan setiap alternatif memiliki solusi ideal yang bersifat positif dan negatif. Untuk segi geografis, penghitungan alternatif dapat menggunakan jarak Euclidian untuk mendapatkan solusi yang terbaik (Rosi, 2016). Metode TOPSIS banyak digunakan dikarenakan memiliki penghitungan yang tidak kompleks dan mudah untuk dimengerti. Tidak hanya itu, komputasi dari metode ini lebih efisien dan kinerja dari tiap alternatif lebih relatif (Gunawan, 2014).

2.3.2 Langkah-Langkah TOPSIS

Dalam melakukan penghitungan TOPSIS terdapat Langkah-langkah yang harus dilakukan. Langkah tersebut yaitu: (Gunawan, 2014)

1. Membangun *normalized decision matrix*

r_{ij} merupakan nilai normalisasi *matrix R* menggunakan metode *Euclidean*, sesuai dengan dibawah ini.

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

dimana:

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


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

2. *Weighted normalized decision matrix*

Bobot (w_1, w_2, \dots, w_n), sesuai dengan persamaan berikut ini.

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

dimana:

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

3. Menentukan solusi ideal positif dilambangkan dengan simbol A^+ dan solusi ideal negatif dilambangkan dengan simbol A^- , sesuai dengan persamaan berikut ini.

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

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

dimana:

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

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

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

4. Menghitung separasi

Yaitu alat ukur antara suatu jarak alternatif ke solusi tiap idealnya, sesuai dengan berikut ini.

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad (2.5)$$

dengan $i = 1, 2, 3, \dots, n;$



$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \quad (2.6)$$

dengan $i = 1, 2, 3, \dots, n$;

5. Menghitung kedekatan kedekatan dari alternatif A^+ dengan solusi ideal A^- , sesuai dengan dibawah ini.

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

6. Merangking alternatif

Perangking alternatif atau pengurutan berdasarkan nilai C_i^+ dari nilai yang paling besar sampai paling kecil. Dengan begitu, alternatif terbaik yaitu alternatif yang berjarak dekat dari solusi ideal positif dan berjarak jauh dari solusi ideal negatif. Alternatif yang memiliki nilai C_i^+ paling besar adalah solusi unggul.

2.4 Pengujian Perangkat Lunak

Dilakukan pengujian sebagai penyelidikan pada perangkat lunak atau aplikasi yang dikembangkan untuk mendapatkan masalah dan mengetahui apakah aplikasi tersebut siap untuk digunakan. Pengujian ini dilakukan juga untuk mencari kesalahan-kesalahan dan juga sebagai evaluasi dari setiap aspek pada sistem maupun fasilitas yang dikembangkan pada perangkat lunak tersebut (Nurhayati, 2020).

Menurut Glen Myers, pengadaan pengujian perangkat lunak memiliki beberapa aturan, yaitu: (Wibisono, 2002)

1. Proses pengujian berupa eksekusi untuk mencari kesalahan dari program tersebut.
2. Terdapat banyak kasus dari kesalahan program dapat dikatakan pengujian dari program tersebut termasuk baik.
3. Dengan adanya kesalahan dari program, pengujian tersebut dikatakan berhasil.

Metode pengujian yang digunakan untuk melakukan evaluasi perangkat lunak yang dikembangkan yaitu pengujian *black box* dan *system usability scale*.

2.4.1 Pengujian *Black-Box*

Merupakan pengujian fungsional terdapat di perangkat lunak dimana melihat kebenaran dari hasil keluaran program sesuai dengan kondisi masukkan tanpa melihat di balik dari proses dari yang bergerak. Pada keluaran



tersebut, pengujian dapat mengukur dan juga mengetahui program yang dikembangkan apakah sudah sesuai dengan pengguna yang membutuhkan. Dengan pengujian ini, diketahui kesalahan-kesalahan yang ada pada program yang dikembangkan (Nurhayati, 2010). Dengan pengujian black box, dapat ditemukan beberapa jenis kesalahan dari program yang dapat diidentifikasi:

- Fungsi yang diberikan tidak ada dalam sistem.
- Tampilan antarmuka yang tidak sesuai.
- Struktur data yang tidak sesuai dengan prosesnya.
- Permasalahan pada mulai maupun akhir dari program.
- Peforma yang tidak seharusnya.

2.4.2 System Usability Scale

System Usability Scale adalah pengujian yang sering digunakan dan juga populer dalam mengukur usability dari perangkat perangkat yang ada terutama apada perangkat bergerak (Baumgartner, 2019). Metode pengujian ini juga memberikan keluasaan kepada pengguna untuk melakukan eksplorasi pada perangkat yang sedang diujikan. Pengguna yang dilakukan uji tersebut diamati untuk nantinya dilakukan evaluasi dan dilakukan analisis.

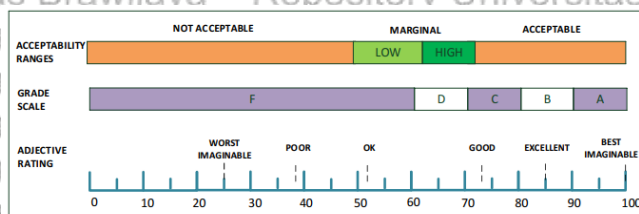
Ada beberapa aspek yang mempengaruhi dari usability sebuah sistem, yaitu *efficiency*, *effectiveness*, *learnability*, *memorability* dan *satisfaction* (Sari, 2019). *Efficiency* yaitu bagaimana sistem tersebut dapat mudah untuk digunakan oleh pengguna. *Effectiveness* yaitu bagaimana ketepatan dari pengguna saat melakukan sebuah pengujian sampai tujuan yang diinginkan. *Learnability* yaitu seberapa mudah pengguna dapat mempelajari sistem tersebut. *Memorability* yaitu bagaimana pengguna dapat mengingat kembali sistem yang telah digunakan tersebut. Terakhir adalah *satisfaction* yaitu kepuasan dari pengguna saat menggunakan sistem tersebut.

Pada pengujian ini, aplikasi diuji tanpa melihat kegiatan atau aktifitas dibelakang dan hanya melihat dari hasil tersebut. Aplikasi dilakukan uji dengan masukkan atau inputan dimana hasil dari pengujian tersebut hanya dari keluaran perangkat lunak tersebut tanpa melihat bagaimana proses dibelakangnya.

Untuk melakukan pengujian dengan metode ini, responden tersebut terdiri dari 5 orang yang dimana akan diberikan kuesioner dimana akan dihitung sesuai cara penghitungan dengan metode SUS. Jumlah dari pertanyaan pada pengujian ini yaitu 10 pertanyaan dimana penghitungan tersebut didapatkan dengan mengurangi setiap nilai skala dari pertanyaan ganjil dengan nilai satu, lalu untuk pertanyaan dengan urutan genap yaitu lima dikurangkan nilai dari jawaban skala pengguna dikurangkan, setelah didapatkan nilai dari masing masing pertanyaan dilakukan perkalian dengan 2,5 dan dijumlahkan semua untuk dilakukan pencarian rata rata sesuai dengan jumlah responden dimana bernilai lima.



Untuk menentukan *grade* dari SUS ada dua acara (Ependi, 2017), yaitu dari sisi pengguna dan dari sisi skala tingkat *grade*. Dari sisi pengguna, terdapat tingkatan yaitu *not acceptable*, *marginal* dan *acceptable*. Dari sisi skala, tingkatannya adalah A, B, C, D dan F. Penentuan hasil dapat dilihat di Gambar 2.1.



Gambar 2.1 System Usability Scale

(Sumber Ependi, 2017)

Range untuk *grade* A adalah lebih besar dari 80,3, *grade* B 74 sampai dibawah 80,3, *grade* C 68 sampai dibawah 74, *grade* D 51 sampai dibawah 68 dan *grade* F dibawah 51. Untuk mendapatkan nilai usabilitas dari sistem yang baik, langkah baiknya nilai dari SUS tersebut lebih dari 68 (MeasuringU, 2020).

2.5 Global Positioning System (GPS)

GPS adalah penentuan posisi sebuah objek dimana menggunakan sistem satelit untuk menentukan posisi tersebut. GPS tersebut digunakan oleh peneliti untuk memberikan letak dari hasil rekomendasi oleh oleh yang didapatkan.

GPS itu sendiri memiliki kurang lebih 24 satelit dimana dapat berfungsi di berbagai cuaca tanpa harus megelluarkan biaya. Satelit GPS mengelilingi bumi dua kali sehari dan mengirimkan sinyal yang tidak biasa kepada perangkat atau device untuk dipecahkan kode tersebut dan mendapatkan jarak dari pengguna (Garmjin, 2019).

2.6 Android

Android, sistem operasi dimana pada dua dekade ini dikembangkan oleh Google digunakan oleh piranti bergerak berlayar sentuh seperti *smartphone* dan *tablet*. Perangkat Bergerak android sudah banyak dimiliki karena harga yang relatif murah dan spesifikasi yang mumpuni (Firman, 2019). Penguji menggunakan perangkat Android untuk pengujian dan pengembangan dengan spesifikasi minimal Android 4.1.

Android sebagai open source berbasis Linux sudah banyak digunakan oleh pengguna maupun developer dapat mengubah data didalam android tersebut secara Cuma Cuma. Beberapa struktur pada android adalah sebagai berikut (Developers, 2018):

- a. Kernel Linux adalah sebuah pondasi dari android untuk melakukan seperti *threading* dan *low-level memory management*.



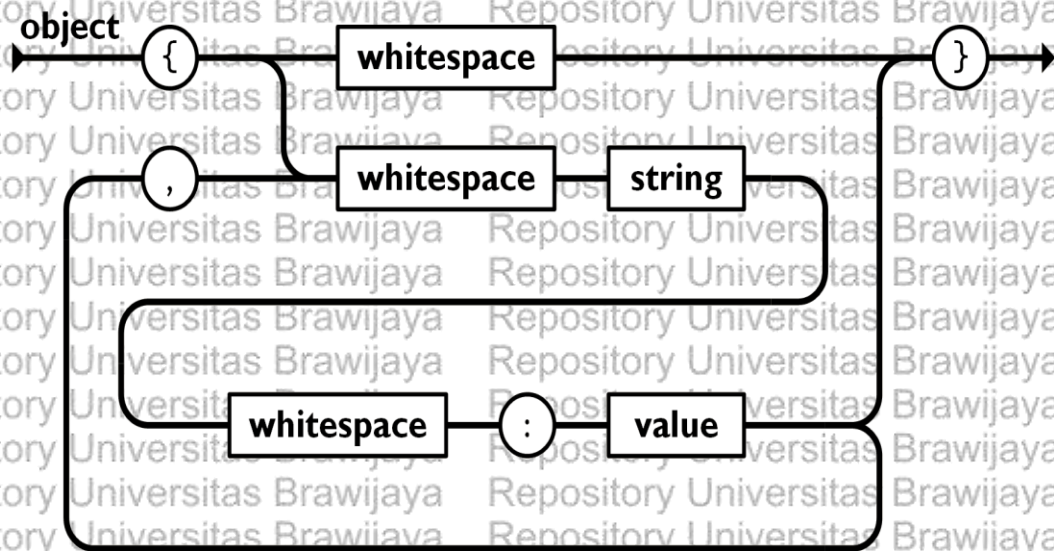
- b. *Java Api Framework* adalah sebuah framework dari Android yang berbasis java dimana developer dapat menggunakannya sesuai dengan kebutuhan.

2.7 Javascript Object Notation (JSON)

JSON, format teks pada berbagai bahasa pemrograman untuk melakukan pertukaran data salah satunya yaitu Java Programming (Crockford, 2015). JSON sendiri merupakan sebuah format teks yang ringan untuk dijalankan dan dilakukan *parsing* antara suatu tempat ke tempat lain. JSON memiliki dua format diantaranya berupa satuan dari nama atau nilai dan berupa daftar nilai yang telah ditata seperti *array*, *sequence*, *list* atau *vector*.

JSON memiliki beberapa format, format tersebut adalah sebagai berikut:

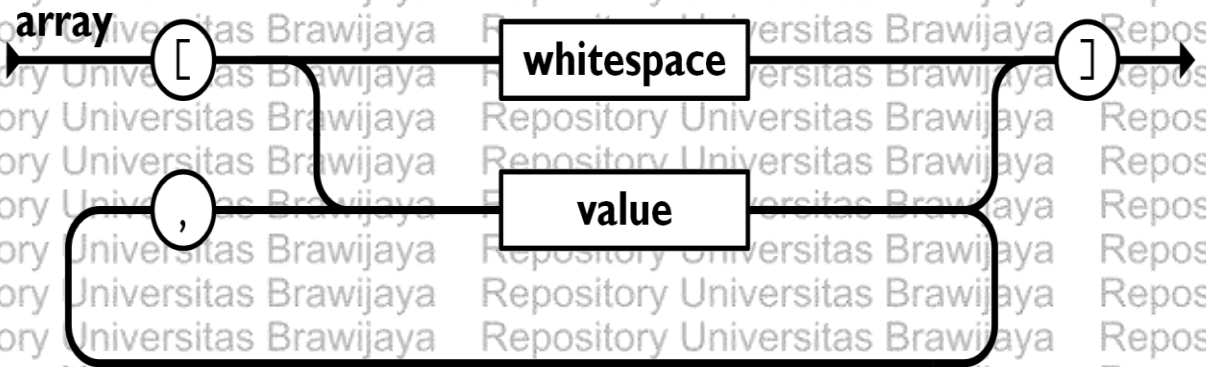
- a. Disebut sebagai JSON objek dimana form ini dimulai dengan { ' dan diakhiri oleh '}'. Untuk nama, dilanjutkan oleh ':' (titik dua) dan untuk *value* yang berbeda beda dibedakan menggunakan ',' (koma). Gambaran objek tersebut pada Gambar 2.4.



Gambar 2.2 Struktur Objek Dalam Format JSON

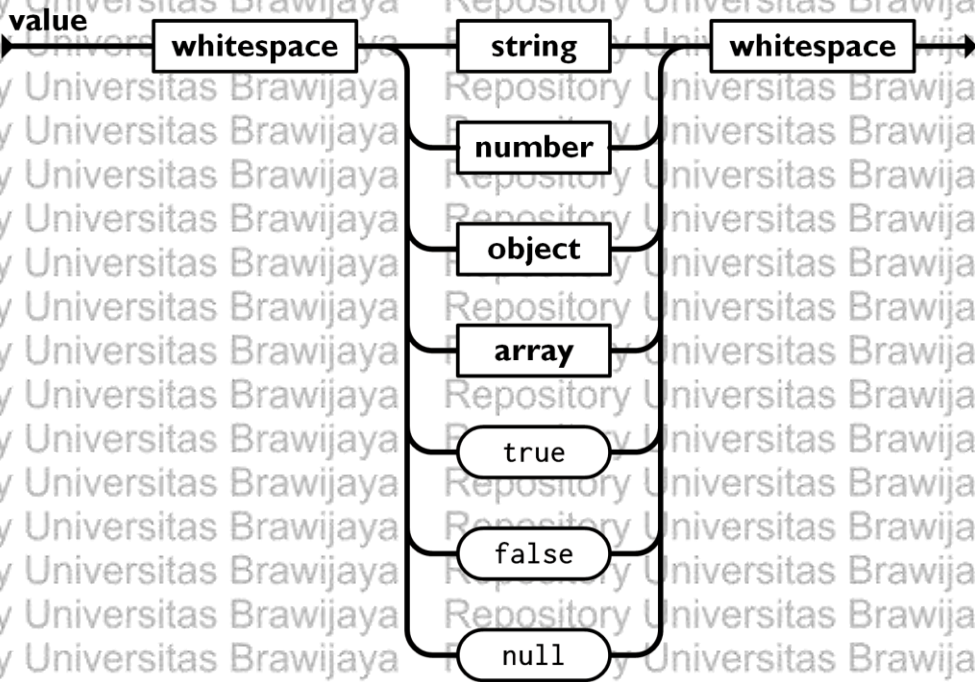
(Sumber Crockford, 2015)

- b. Disebut sebagai JSON array dimana form ini dimulai dengan [' dan diakhiri oleh ']' Untuk nilai yang berbeda beda dibedakan menggunakan ',' (koma). Gambaran objek tersebut pada Gambar 2.5.



Gambar 2.3 Struktur Array Dalam Format JSON
(Sumber Crockford, 2015)

c. Nilai dari JSON tersebut dapat berupa angka, *double quotes*, atau *condition value* atau objek maupun array dimana disebut sebagai *nested*. Gambaran terlihat di Gambar 2.6.



Gambar 2.4 Struktur Nilai Dalam Format JSON
(Sumber Crockford, 2015)

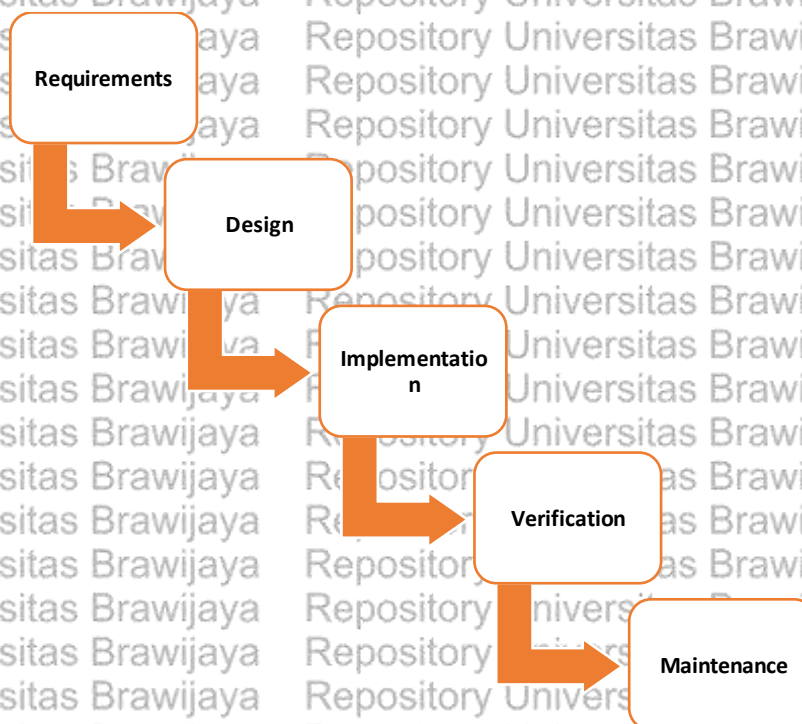
2.8 Waterfall

Waterfall adalah model SDLC yang dapat berperan saat melakukan sebuah pengembangan sistem. Model *waterfall* pertama kali di populerkan oleh Winston W. Royce pada tahun 1997 dengan menjelaskan modelnya berjalan secara simetris dimana sebuah proyek atau penelitian harus direncanakan secara matang dan



dieksekusi secara baik (Kramer, 2018). Proses dari model SDLC ini dimulai dari *requirement, design, implementation, verification dan maintenance*. Deskripsi dari proses pada model *waterfall* dapat dilihat dibawah ini:

1. **Requirement** : Dilakukan pendeskripsian kebutuhan dari sistem yang akan dibuat. Deskripsi kebutuhan tersebut terdapat fitur-fitur dari sistem dan spesifikasi sistem yang akan dikembangkan.
2. **Design** : Melakukan desain dari algoritme, basis data, arsitektur desain dari sistem, logic dari sistem dan struktur yang didefinisikan dari sistem. Tahap ini dilakukan setelah informasi-informasi dari tahap *requirement* telah selesai dirumuskan.
3. **Implementation** : Setelah tahap *requirement* dan *design* telah selesai dilakukan, dilakukan implementasi dengan basis pengerjaan sesuai dengan tahapan sebelumnya.
4. **Verification** : Tahap ini yaitu melakukan verifikasi berupa testing pada sistem yang sudah di-implementasikan dan melakukan *bug-test* dan *bug-fix* sampai sistem yang telah dibuat *bug-free*.
5. **Maintenance** : Jika adanya tambahan fitur, dilakukan penambahan fitur-fitur baru pada tahap ini, dan juga dilakukannya pemeliharaan jika ada *error* kecil yang tidak ditemukan pada tahapan sebelumnya.



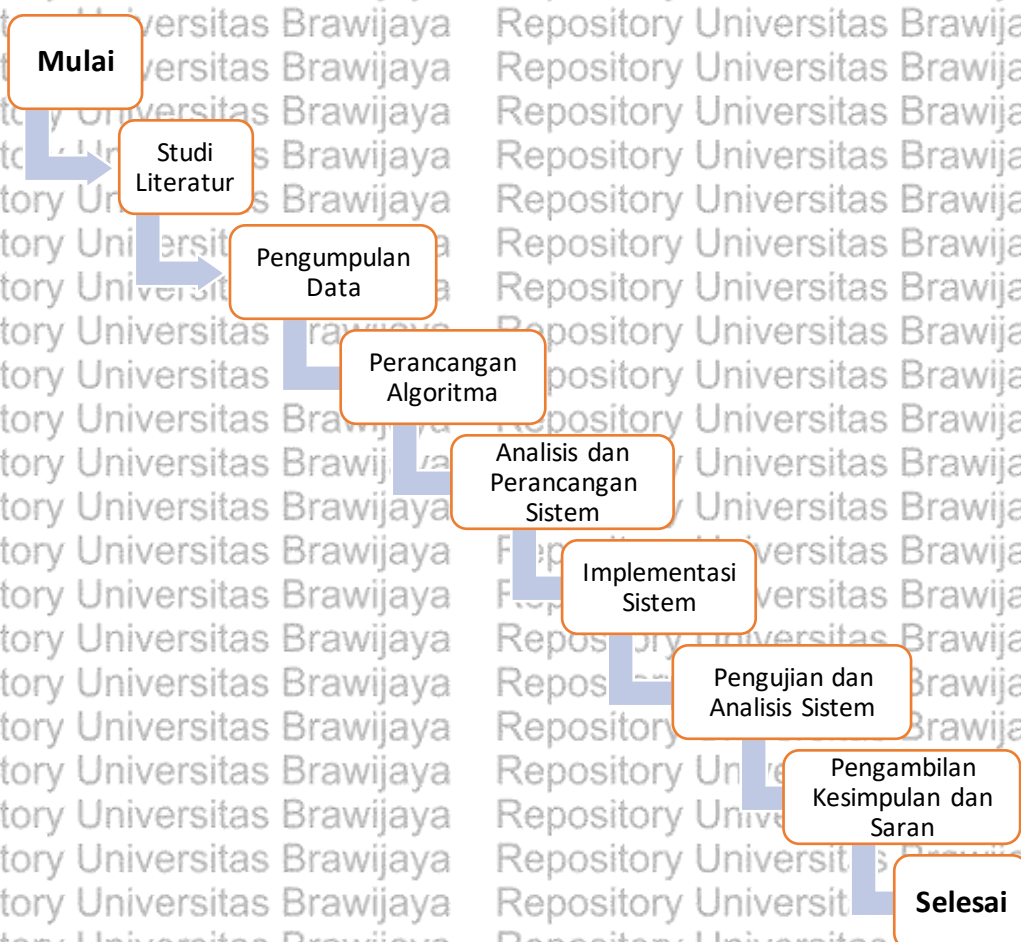
Gambar 2.5 Metode Waterfall

(Sumber Kramer, 2018)



BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan bagaimana langkah yang dikerjakan dan metode untuk melakukan penelitian. Gambar 3.1 menggambarkan bagaimana alur penelitian berlangsung.



Gambar 3.1 Alur Metodologi Penelitian

3.1 Metode Penelitian

Alur ditampilkan Gambar 3.1 memperlihatkan tahap pengerjaan sistem rekomendasi tempat oleh-oleh dimulai dari studi literatur sampai Pengambilan kesimpulan dan saran. Diawali dengan studi literatur, dimana studi literatur sebagai tahap pemahaman konsep dari teori yang merujuk kepada pengembangan yang akan dibuat. Setelah Studi literatur, tahap berikutnya adalah pengumpulan data. Dikumpulkan data pada penelitian tempat oleh-oleh camilan di kota malang, yaitu nama oleh-oleh, harga, maupun letak dari tempat oleh-oleh tersebut. Pada bab 2 dijelaskan bahwa SDLC yang digunakan adalah *waterfall*, dimana pada



diagram alur peran studi literatur dengan pengumpulan data termasuk pada tahap *requirement* pada SDLC *waterfall*.

Tahap berikutnya adalah melakukan analisis kebutuhan untuk sistem yang akan dibuat. Lalu setelah semua kebutuh telah ditentukan, merancang sebuah sistem menjadi arsitektur arsitektur sistem dimana terdapat model model perancangan perangkat bergerak, diagram-digram, *database*, perancangan algoritme dan desain *interface*. Jika tahap tersebut telah dilaksanakan, lalu melakukan implementasikan hasil analisis dan perancangan yang sudah dikemukakan.

Tahap selanjutnya adalah pengujian terhadap sistem tersebut. Pengujian sistem itu sendiri bertujuan untuk mengetahui apakah sebuah sistem memiliki fungsional yang mumpuni dan juga melakukan validasi dari algoritme tersebut dengan melihat hasil yang stabil jika ada kriteria yang berbeda. Setelah pengujian tahap pengujian selesai, dilakukan analisis dari pengujian tersebut apakah sudah sesuai dengan tujuan dari penelitian ini dan hasilnya sudah benar. Tahap yang terakhir adalah pengambilan kesimpulan dan saran, dimana kesimpulan berisi hasil yang digunakan untuk melakukan penjawaban dari rumusan masalah yang telah dirumuskan. Dan saran berisi perbaikan perbaikan kesalah dari peneliti agar dapat disempurnakan lebih baik lagi untuk penelitian berikutnya.

3.2 Tipe Penelitian

Penelitian dari pengembangan rekomendasi oleh-oleh ini merupakan tipe implementatif yang termasuk dalam kategori pengembangan. Pengembangan ini menghasilkan sebuah perangkat lunak atau program yang didasarkan dari solusi yang didapatkan dari sebuah permasalahan. Perangkat lunak tersebut dikembangkan sesuai dengan solusi dari permasalahan yang sesuai dengan analisis dan perancangan yang selanjutnya akan di-implementasi agar sesuai dengan kebutuhan yang ada dan diuji secara real agar menjadi sebuah hasil yang utuh.

3.3 Subjek, Lokasi dan Data Penelitian

Oleh-oleh menjadi subjek dari penelitian dimana oleh-oleh dibatasi hanya camilan saja. Lokasi dari penelitian ini mencakup Kota Malang, Jawa Timur. Data dari tempat oleh-oleh yang dibutuhkan adalah nama dari tempat oleh-oleh, lokasi tempat oleh-oleh, nama oleh-oleh dari tiap tempat oleh-oleh begitu juga dengan harga tiap oleh-oleh. Peneliti melakukan observasi pada proses pencarian data oleh-oleh.

3.4 Metode Pengumpulan Data

Pengumpulan informasi untuk tempat oleh-oleh khas Malang peneliti dapatkandari observasi secara langsung dan internet. Data yang kumpulkan seperti nama tempat, lokasi tempat, barang yang dijual maupun harga yang dicantumkan dan lamanya kadaluarsa untuk makanan ringan. Untuk kebutuhan



yang digunakan, pengumpulan data melalui kuesioner dengan jumlah responden yaitu 40 orang dengan umur diatas 17 tahun dan sering membelanjakan oleh-oleh untuk kerabatnya.

3.5 Peralatan Pendukung

Piranti keras untuk melakukan pengembangan pengembangan berupa sebuah laptop. Performansi yang cukup meliputi:

Sistem Operasi : Windows 10 Pro 64-bit

Prosesor : Intel Core i7-4710HQ @ 2.50Ghz

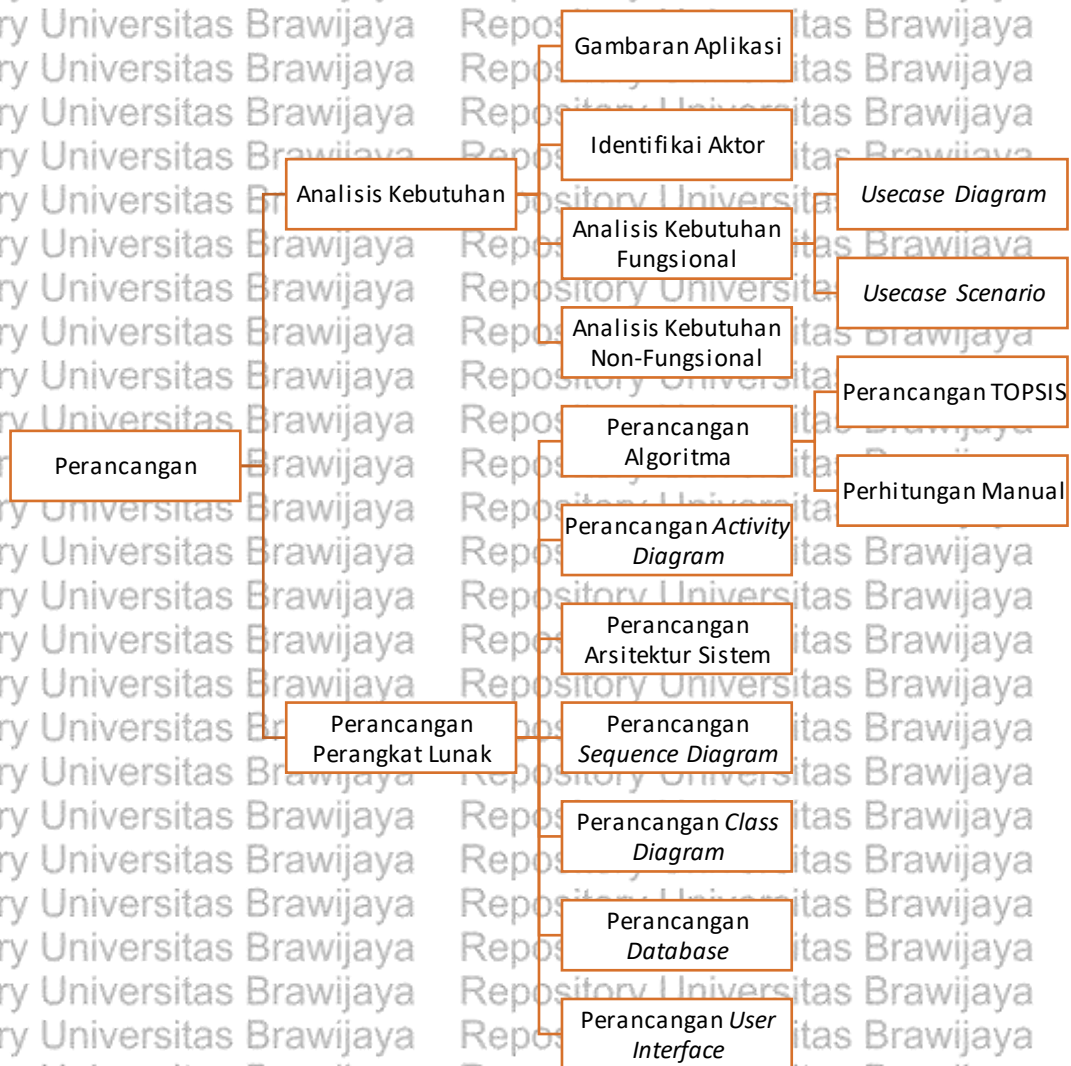
Memory : 8192MB

3.6 Metode Pengembangan Sistem

Mengembangkan suatu system butuh sebuah metode pengembangan system. Metode pengembangan system itu sendiri adalah teknik dalam melakukan pengembangan system yang akan dibuat. Pengembangan rekomendasi tempat oleh-oleh, menggunakan metode perangkat lunak yaitu *waterfall*. Metode tersebut dilihat tahapannya di Gambar 2.2.

BAB 4 ANALISIS DAN PERANCANGAN

Penjelasan bab ini tergambar di Gambar 4.1. Akan dijelaskan apa saja analisis yang didapatkan dan bagaimana perangkat bergerak dirancang.



Gambar 4.1 Diagram Pohon Analisis Dan Perancangan

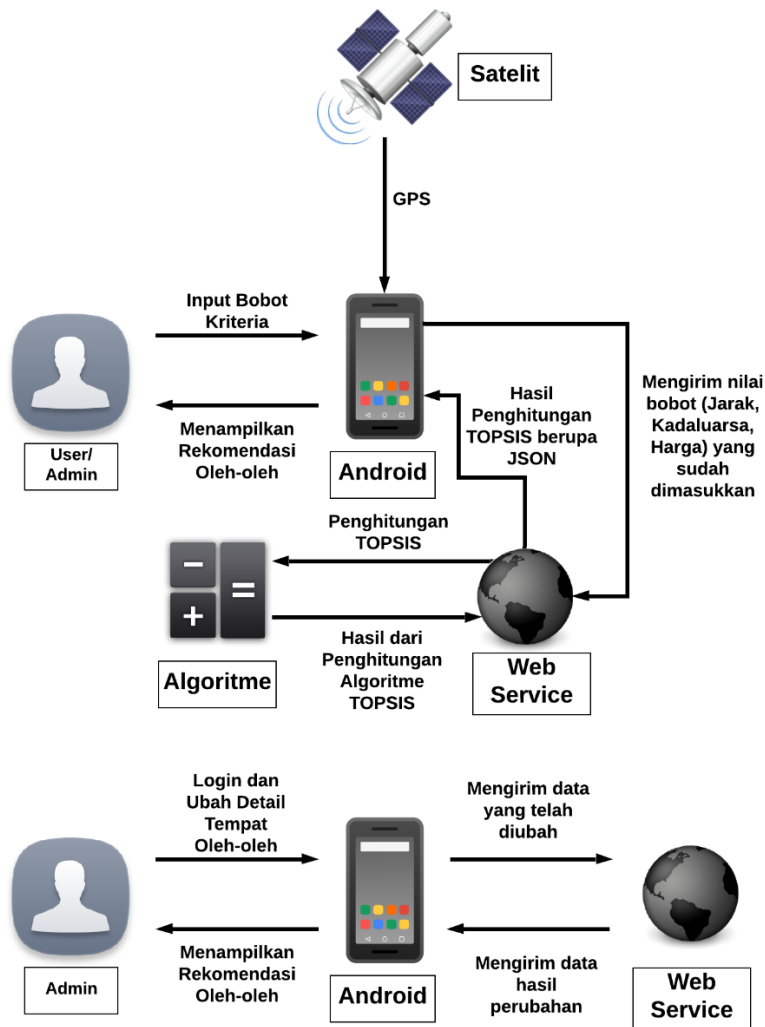


4.1 Analisis Kebutuhan

Penjelasan berisikan identifikasi kebutuhan apa saja yang ada dari perangkat bergerak rekomendasi oleh-oleh khas Malang. Tahap dari analisis ini: menggambarkan bagaimana pengembangan akan dikembangkan, mengidentifikasi siapa saja aktor yang menggunakannya, melakukan analisis dari kebutuhan yang diperlukan secara fungsionalitas, analisis use case diagram dan scenario dan analisis kebutuhan non fungsional. Tujuan tersebut untuk memberikan sketsa tentang apasaja persiapan untuk sebuah sistem sesuai dengan kebutuhan.

4.1.1 Gambaran Aplikasi

Gambar dibawah ini merepresentasikan bagaimana system akan berjalan. Penggambaran aplikasi merupakan tahap awal dalam mengembangkan sebuah sistem.



Gambar 4.2 Gambaran Aplikasi Rekomendasi Oleh-oleh



Pada penggambaran aplikasi diatas, pengguna dan admin pertama kali menginputkan bobot kriteria yang terdiri dari harga , jarak dan kadaluarsa yang diinginkan oleh pengguna. Sebelum aplikasi tersebut diluncurkan, aplikasi akan meminta akses *GPS* pada perangkat pengguna. Nilai bobot kriteria tersebut di-*input*-kan oleh pengguna dan koordinat lokasi dikirim menggunakan *web service*. Data yang dikirim akan diterima oleh method algoritme perhitungan TOPSIS. Method tersebut akan memproses data yang telah diterima. Setelah diproses, method menghasilkan rekomendasi tempat oleh-oleh dengan format JSON dimana hasil tersebut akan ditampilkan ke aplikasi pengguna.

Pada menu awal, terdapat juga pilihan admin dimana terdapat pilihan untuk melakukan perubahan atas lama kadaluarsa, harga paling murah dan menu menu yang disediakan. Admin merupakan aktor yang melakukan perubahan di masing masing tempat oleh-oleh untuk mempermudah pengubahan data dari masing masing tempat oleh-oleh. Admin disini merupakan user pada tiap tempat oleh-oleh dimana terdapat *username* dan *password* sesuai dengan yang sudah diset untuk tiap tempat tersebut.

4.1.2 Identifikasi Aktor

Tujuan dari identifikasian aktor ini untuk mengetahui hubungan antar *actor* dengan pengembangan ini. Penjelasan terdapat di Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna memasukkan bobot dari kriteria harga, jarak dan kadaluarsa. Pengguna juga mendapatkan hasil rekomendasi oleh-oleh dan melihat detail dari hasil rekomendasi tersebut.
Admin	Admin merupakan user dimana tiap tempat oleh-oleh mempunyai <i>username</i> dan <i>password</i> yang berbeda. Admin melakukan perubahan atas data dari harga, kadaluarsa dan menu yang ditampilkan pada rekomendasi tersebut.

4.1.3 Analisis Kebutuhan Fungsional

Terdapat fungsionalitas sebuah kebutuh dan *use case* dipaparkan dari kebutuhan fungsional tersebut yang akan dirincikan. Kebutuhan fungsional tersebut dapat dilihat di Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional

Identifier	Kebutuhan Fungsional	Nama Use Case
ROF-01	Sistem dapat menerima nilai bobot dari setiap kriteria yang ada.	Memasukan bobot kriteria oleh-oleh

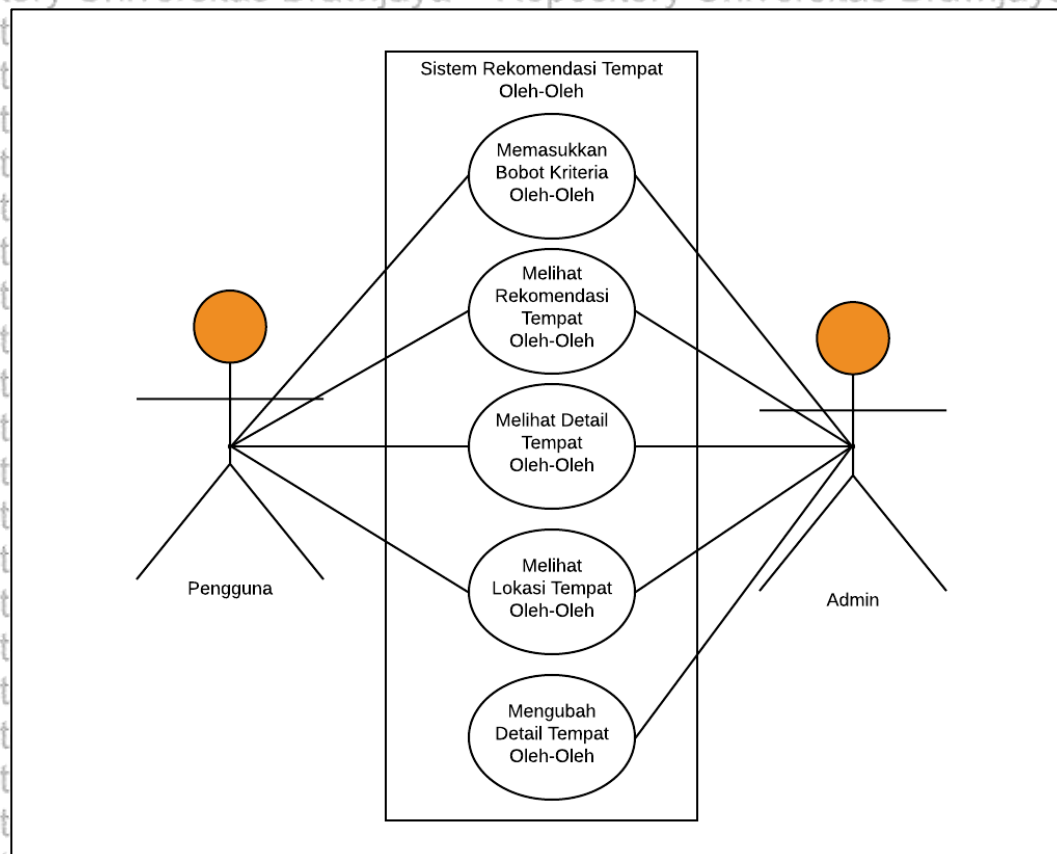


ROF-02	Sistem dapat menampilkan rekomendasi tempat oleh-oleh.	Melihat rekomendasi tempat oleh-oleh
ROF-03	Sistem dapat menampilkan detail tempat oleh-oleh.	Melihat detail tempat oleh-oleh
ROF-04	Sistem dapat menampilkan lokasi tempat oleh-oleh.	Melihat lokasi tempat oleh-oleh
ROF-05	Sistem dapat melakukan perubahan atas input.	Mengubah detail tempat oleh-oleh

Penjabaran lebih lanjutnya akan dijelaskan lebih spesifik menggunakan diagram dibawah ini.

4.1.3.1 Use Case Diagram

Diagram ini adalah penggambaran dari perlakuan aktor pada sistem yang dikembangkan dengan representasi dalam sebuah diagram. Digram dari rekomendasi menyesuaikan dengan fungsional dari sistem tersebut. Penggambaran dari *use case* tersebut dapat dilihat di Gambar 4.3 berikut ini.



Gambar 4.3 Use Case Diagram Kebutuhan Fungsional

4.1.3.2 Use Case Scenario

Komponen dari tersebut terdiri: aktor yang terlibat, tujuan penggunaan scenario tersebut, deskripsi scenario tersebut, langkah-langkah *Use Case*



Scenario, keadaan pertama kali yang harus dilakukan (*pre-condition*), keadaan yang terakhir sebagai hasil dari proses yang telah dilakukan pada aliran utama (*post-condition*), dan aliran yang menjadikan opsi lain jika tidak memenuhi *main flow* disebut *alternatif flow* dan aliran utama (*main flow*) sebagai jalannya proses. *Use Case Scenario* tersebut terdapat di Tabel 4.3 - Tabel 4.7.

Tabel 4.3 *Use Case Scenario* Memasukkan Bobot Kriteria Oleh-oleh

Nama	Uraian
Aktor	Pengguna
Tujuan	Memasukkan nilai dari bobot kriteria oleh-oleh (Harga, jarak, kadaluarsa)
Deskripsi	Melakukan pengisian nilai dari 1 – 100 pada tiap bobot kriteria tersebut.
Pre-condition	Aktor membuka aplikasi rekomendasi oleh-oleh, lalu akan muncul halaman dimana terdapat bobot kriteria.
Main Flow	Aktor memasukkan nilai bobot dari tiap kriteria dengan range dari bobot tersebut dari 1-100.
Alternative Flow	Aplikasi akan menampilkan <i>error</i> jika nilai dari bobot kriteria bernilai 0 atau tidak diisi.
Post-condition	Aktor menekan tombol <i>next</i> setelah nilai bobot tiap kriteria telah diisi.

Tabel 4.4 *Use Case Scenario* Melihat Rekomendasi Tempat Oleh-oleh

Nama	Melihat Ringkasan
Aktor	Pengguna
Tujuan	Untuk menampilkan rekomendasi tempat oleh-oleh dari masukan pengguna.
Deskripsi	Menampilkan tiga rekomendasi yang menyesuaikan inputan pengguna.
Pre-condition	Aktor mengisi nilai bobot dan menekan tombol <i>next</i> pada halaman yang berisi bobot dari kriteria rekomendasi oleh-oleh
Main Flow	Aplikasi menampilkan hasil rekomendasi tempat oleh-oleh sesuai dengan bobot dari kriteria yang pengguna masukkan.
Post-condition	Aktor memilih rekomendasi tempat oleh-oleh yang tersedia.



Tabel 4.5 Use Case Scenario Melihat Detail Tempat Oleh-oleh

Nama Use Case	Melihat Hasil Rekomendasi
Aktor	Pengguna
Tujuan	Untuk menampilkan detail dari tempat oleh-oleh.
Deskripsi	Menampilkan detail tempat oleh-oleh yang dipilih oleh pengguna.
Pre-condition	Aktor memilih salah satu hasil rekomendasi tempat oleh-oleh
Main Flow	Aplikasi menampilkan detail tempat rekomendasi oleh-oleh.
Post-condition	Aktor menekan tombol <i>Maps</i> .

Tabel 4.6 Skenario Use Case Melihat Lokasi Tempat Oleh-oleh

Nama Use Case	Melihat Hasil Rekomendasi
Aktor	Pengguna
Tujuan	Untuk menampilkan lokasi dari tempat oleh-oleh.
Deskripsi	Menamplkan lokasi tempat oleh-oleh sesuai dengan pemilihan <i>user</i> .
Pre-condition	Aktor menekan tombol <i>Maps</i> .
Main Flow	Aplikasi menampilkan lokasi dari tempat rekomendasi oleh-oleh.
Post-condition	

Tabel 4.7 Skenario Use Case Mengubah Detail Tempat Oleh-oleh

Nama Use Case	Melihat Hasil Rekomendasi
Aktor	Admin
Tujuan	Untuk melakukan perubahan atas detail dari tempat oleh-oleh
Deskripsi	Menamplkan nama tempat, harga paling murah, kadaluarsa tercepat dan kondisi tempat tersebut buka atau tidak dimana dapat diubah sesuai dengan tempat oleh-oleh tersebut.



Pre-condition	Aktor telah login sesuai dengan <i>username</i> dan <i>password</i> masing-masing pada menu admin.
Main Flow	Aplikasi dapat melakukan perubahan atas detail tempat dari oleh-oleh tersebut.
Post-condition	Aktor memilih tombol Simpan.

4.1.4 Analisis Kebutuhan Non-Fungsional

Untuk memenuhi kelengkapan dari pengembangan ini, adanya kebutuhan non-fungsional adalah salah satunya. Kebutuhan ini tidak *user* inginkan, tetapi dengan adanya kebutuhan ini, *user* mendapatkan aplikasi tersebut menjadi lebih kompeten. Kebutuhan tersebut dapat dilihat pada Tabel 4.8.

Tabel 4.8 Spesifikasi Kebutuhan Non Fungsional

Identifler	Parameter	Deskripsi
ROFN-01	Validasi Algoritme	Melakukan validasi terhadap keluaran sistem dengan manualisasi hitung.
ROFN-02	<i>System Usability Scale</i>	Melakukan tes usability terhadap perangkat bergerak rekomendasi oleh-oleh.

4.2 Perancangan Perangkat Lunak

Perancangan ini membahas melibatkan kebagaimanaan aplikasi dikembangkan. Untuk melakukan pengembangan ini terdapat beberapa prosedur, diantaranya: perancangan algoritme, perancangan diagram seperti *activity*, *sequence* dan lainnya, perancangan penyimpanan system secara *database*, dan perancangan penampilan pada aplikasi disebut dengan anatar muka.

Selain itu, terdapat juga aritektur system yang dirancang sebagaimana berisikan sebuah sistem bergerak dari mulai aplikasi itu sendiri sampai bagaimana pengeluarannya dengan beberapa penjelasan untuk memberikan gambaran pada perancangan tersebut. Untuk perancangan algoritme itu sendiri terdiri dari tiga tahapan. Tahapan tersebut meliputi: perancangan Haversine, perancangan TOPSIS dan perhitungan manual. Terakhir yaitu perancangan antarmuka dimana terdapat flow dari antarmuka dari pengembangan aplikasi ini.

4.2.1 Perancangan Algoritme

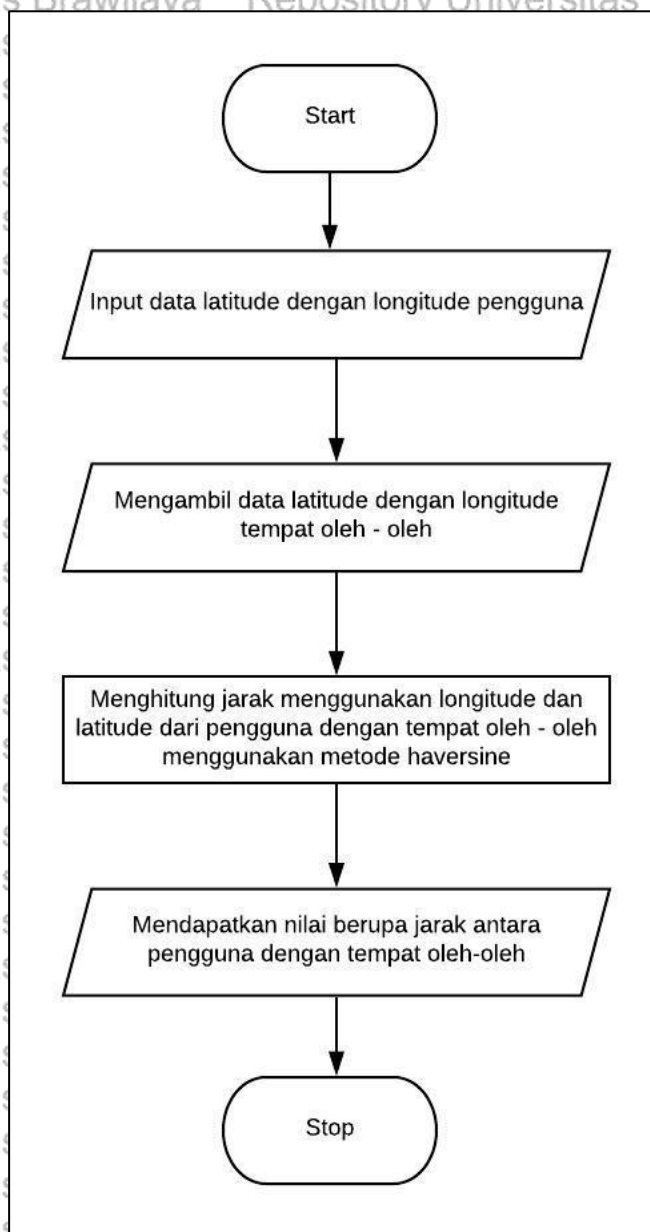
Dijelaskan pada sub Perancangan algoritme apa saja metode algoritme yang digunakan untuk menempuh pengembangan rekomendasi. Kriteria TOPSIS pada rekomendasi oleh-oleh ini yaitu Jarak, Harga dan Kadaluarsa didapatkan dari resopenden survei yang terdiri dari 40 responden dimana 75% dari responden tersebut sering membelikan oleh-oleh untuk keluarga maupun kerabat. Dalam survei tersebut diketahui bahwa jarak, harga maupun kadaluarsa menjadi hal penting untuk membeli sebuah oleh-oleh. Perancangan algoritme terdiri dari



perancangan haversine, perancangan TOPSIS begitu juga bagaimana metode tersebut terhitung secara manual.

4.2.1.1 Haversine

Haversine merupakan metode untuk mencari *longitude* dengan *latitude* dari dua tempat yang berbeda. Langkah dari Haversine terdapat pada Gambar 4.4. Untuk contoh penghitungan TOPSIS terdapat pada sub bab 4.2.1.3 bagian Perhitungan Manual.



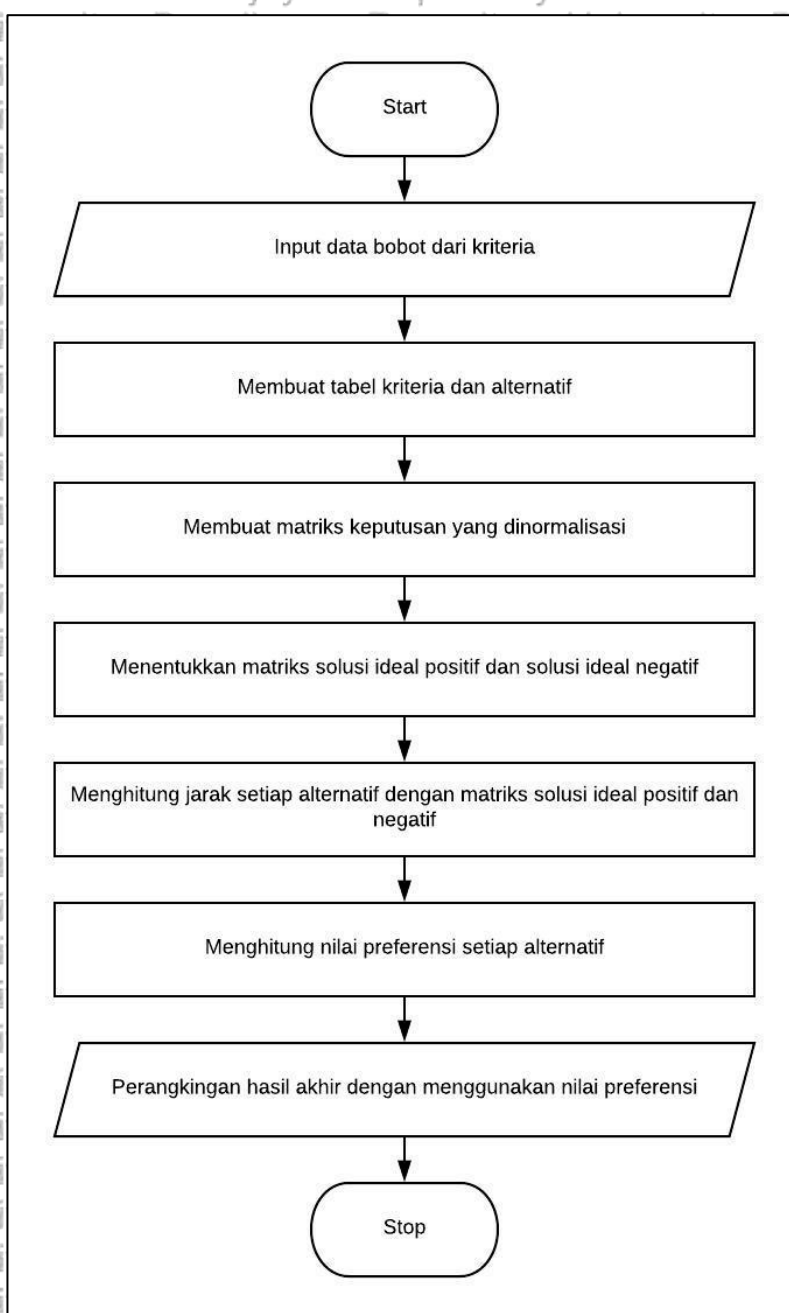
Gambar 4.4 Flow Chart Haversine



4.2.1.2 TOPSIS

TOPSIS merupakan metode dari pengambilan keputusan dimana memberikan nilai hasil yang akan ditampilkan oleh rekomendasi oleh-oleh berdasarkan kriteria yang terpilih dengan parameter yang ditentukan oleh pengguna.

Langkah dari TOPSIS digambarkan pada Gambar 4.5.



Gambar 4.5 Flow Chart TOPSIS



4.2.1.3 Perhitungan Manual

Dari metode untuk tiap algoritme yang telah dijelaskan sebelumnya, terdapat contoh untuk melakukan penghitungan tiap metode yang mendukung aplikasi pengembangan rekomendasi ini. Untuk perhitungannya dapat dilihat sebagai berikut:

- **Rumus Haversine**

Dari perancangan Haversine yang terdapat pada gambar 4.4, penggambaran untuk menghitung metode tersebut dijelaskan secara ringkas. Tabel 4.9 menjelaskan perhitungan dari dua titik yang berbeda dengan *Latitude* dan *Longitude* sebagai data untuk mencari sebuah jarak antar kedua titik tersebut. Nilai R adalah 6371,1 kilometer (km).

Tabel 4.9 Data Lokasi

Lokasi	Latitude	Longitude
Pengguna	-7,947421	112,6243284
Tujuan	-7,940652	112,623967

$$A = \sin^2\left(\frac{(-7,940652) - (-7,947421)}{2}\right)$$

$$B = \sin^2\left(\frac{(112,623967) - (112,6243284)}{2}\right)$$

$$C = A + \cos(-7,947421) \cos(-7,947421) * B$$

$$d = 2 * 6371,1 * \arcsin \sqrt{C}$$

$$d = 0.753730003$$

Didapatkan hasil perhitungan antara 2 titik tempat yang berbeda dimana telah ditentukan dengan hasil 0.753730003 km.

- **Perhitungan TOPSIS**

Perhitungan manual dari TOPSIS terdapat 3 kriteria yaitu : J (Jarak dalam km), K (Kadaluarsa satuan hari) dan H (Harga satuan rupiah). Dari 3 kriteria tersebut, akan dihitung sesuai dengan rumus TOPSIS. Cara untuk perhitungan TOPSIS dapat dilihat pada contoh berikut ini:

- a. Data Alternatif yang akan digunakan

Prihal data yang sebagai contoh alternatif tertampilkan kriteria berupa Jarak, Kadaluarsa dan Harga dimana terdapat pada Tabel 4.10.

Tabel 4.10 Data Sample

No	Alternatif Tempat Oleh-oleh	J	K	H
1	Malang Strudel Sukarno Hatta 2	2,2	3	80000



2	Lapis Malang Kawi	0,98	4	50000
3	Pia Cap Mangkok	4	12	30000
4	Pia Cap Mangkok Semeru	2,5	13	20000
5	Goedang Oleh-oleh	2	10	35000

b. Pencarian normalisasi dari alternative yang dijadikan matriks

Melakukan pencarian normalisasi menggunakan penggunaan rumus terdapat pada Persamaan (2.1) dimana langkah pertama yaitu melakukan penjumlahan dari nilai alternatif dari kriteria yang ada lalu dipangkatkan dua dan diakar kuadratkan. Setelah itu, nilai pada *cell* pada table tersebut dilakukan operasi pembagi dengan nilai hasil tiap kriteria. Nilai dari hasil penghitungan alternatif dan normalisasi dilihat dibawah ini (Tabel 4.11 dan Tabel 4.12).

Tabel 4.11 Hasil Penjumlahan Alternatif Tiap Kriteria

No	Alternatif Tempat Oleh-oleh	J	K	H
1	Malang Strudel Sukarno Hatta 2	4,84	9	6400000000
2	Lapis Malang Kawi	0,9604	16	2500000000
3	Pia Cap Mangkok	16	144	9000000000
4	Pia Cap Mangkok Semeru	6,25	169	4000000000
5	Goedang Oleh-oleh	4	100	12250000000
		10,46138	48,77499	191898,4

Berikut ini adalah contoh hitungan diatas pada Kolom J:

$$\sqrt{4,84^2 + 0,9604^2 + 16^2 + 6,25^2 + 4^2 + \dots} = 10,4613$$

Tabel 4.12 Hasil Normalisasi Matriks Alternatif

No	Alternatif Tempat Oleh-oleh	J	K	H
1	Malang Strudel Sukarno Hatta 2	0,210297	0,061507	0,416887
2	Lapis Malang Kawi	0,093678	0,082009	0,260555
3	Pia Cap Mangkok	0,382359	0,246028	0,156333



4	Pia Cap Mangkok Semeru	0,238974	0,26653	0,104222
5	Goedang Oleh-oleh	0,191179	0,205023	0,182388

Berikut ini adalah perhitungan untuk normalisasi matriks alternatif.

$$\text{- Baris 1, Kolom J: } \frac{2,2}{10,46138} = 0,210297$$

c. Pencarian nilai normalisasi matriks terbobot

Untuk tahap pencarian normalisasi matriks tersebut, sesuai dengan rumus Persamaan (2.2), dimana diawali perkalian dari yang sebelumnya sudah dihitung pada matriks yang tersedia pada Tabel 4.12 dengan nilai bobot yang ditentukan sebelumnya. Nilai pencarian dari matriks tersebut terdapat pada Tabel 4.13.

Tabel 4.13 Hasil Normalisasi Matriks Terbobot

No	Alternatif Tempat Oleh-oleh	J	K	H
1	Malang Strudel Sukarno Hatta 2	16,82379	5,535623	29,18211
2	Lapis Malang Kawi	7,494234	7,380831	18,23882
3	Pia Cap Mangkok	30,58871	22,14249	10,94329
4	Pia Cap Mangkok Semeru	19,11794	23,9877	7,295527
5	Goedang Oleh-oleh	15,29435	18,45208	12,76717

Dibawah ini bagaimana penghitungan normalisasi dari matriks pada table sebelumnya yang dinormalisasikan dengan bobot.

$$\text{- Baris pertama, Kolom J: } 0,210297 * 80 = 16,82379$$

d. Pencarian nilai solusi ideal

Penghitungan dalam melakukan pencarian solusi ideal, sesuai penggunaan rumus pada Persamaan (2.3) dan (2.4) dimana menggunakan nilai matriks dari hasil di Tabel 4.13. Pada penghitungan ini, pencarian nilai yang menguntungkan didapat jika nilai tertinggi dihasilkan pada solusi ideal positif. Sedangkan pemberian penghasilan pada pencarian yang merugikan, memberikan nilai solusi negatif yang tinggi.

Tabel 4.14 Hasil Solusi Ideal Positif dan Solusi Ideal Negatif



No	Solusi	J	K	H
1	A+	30,58870862	36,90416	29,18211
2	A-	5,353024008	5,535623	3,647763

Hasil perhitungan pada Tabel 4.14 membuahkn hasil dimana contoh dari perhitungan tersebut terdapat pada contoh dibawah ini.

- Baris 1, Kolom J:
 $\max\{16,82379;7,494234;30,58871;19,11794;15,29435;...\} = 30,58871$
- Baris 2, Kolom J:
 $\min\{5,535623;7,380831;22,14249;23,9877;18,45208;...\} = 5,353024008$

e. Pencarian nilai jarak pada tiap alternatif terhadap solusi ideal

Pencarian nilai untuk melakukan hitungan jarak alternatif menyesuaikan pada rumus pada Persamaan (2.5) dan Persamaan (2.6) dengan nilai dari penghasilan yang telah dihitung sesuai pada Tabel 4.13 dan nilai dari solusi ideal yang sebelumnya sudah dihitung pada Tabel 4.14. Langkah awal diawali dari pengakaran dari hasil dari pengkuadratan yang sudah dijumlahkan dari normalisasi terbobot pada tiap barisnya dikurangi nilai dari solusi ideal positif dan pengakaran hasil menggunakan pengkuadratan yang sudah dijumlahkan antara nilai dari normalisasi terbobot pada tiap barisnya dikurangi dengan nilai dari solusi ideal negatif. Dari pencarian hasil tersebut, nilai-nilai tersebut dipaparkan pada Tabel 4.15.

Tabel 4.15 Hasil Perhitungan Jarak Terhadap Solusi Ideal Positif dan Negatif

No	Jarak	
	SI+	SI-
1	34,25577	27,99252011
2	39,04788	14,86231531
3	23,46404	31,07816976
4	27,88253	23,30790343
5	29,04904	18,67695008

Salah satu contoh dari perhitungan alternatif jarak terhadap solusi tiap ideal adalah sebagai berikut:

- Baris 1, Kolom SI+:

$$\sqrt{(16,82379 - 30,58870862)^2 + (5,535623 - 36,90416)^2 + (29,18211 - 29,18211)^2} = 34,25577$$

- Baris 1, Kolom SI-:



$$\sqrt{(16,82379 - 5,353024008)^2 + (5,535623 - 5,535623)^2 + (29,18211 - 3,647763)^2} \\ = 27,99252011$$

f. Pencarian hasil penghitungan preferensi

Perhitungan untuk nilai ini menggunakan rumus pada Persamaan (2.7) dengan nilai untuk data yang sebelumnya dihitung pada Tabel 4.15. Hasil penghitungan didapatkan dengan membagikan nilai solusi ideal negatif dengan total nilai dari solusi ideal yaitu negatif dan positif yang dimana telah dijumlahkan. Hasil dari pencarian hitungan tersebut pada Tabel 4.16.

Tabel 4.16 Hasil Perhitungan Nilai Prefrensi

No	Alternatif Tempat Oleh-oleh	Preferensi
1	Malang Strudel Sukarno Hatta 2	0,44969138 2
2	Lapis Malang Kawi	0,27568653
3	Pia Cap Mangkok	0,56980035 1
4	Pia Cap Mangkok Semeru	0.45531753 3
5	Goedang Oleh-oleh	0.39133711 2

Salah satu contoh dari perhitungan alternatif preferensi pada Tabel 4.15 akan dijelaskan pada perhitungan dibawa ini.

$$\text{Baris 1: } \frac{27,99252011}{(34,25577 + 27,99252011)} = 0,449691382$$

g. Perangkingan nilai preferensi

Dengan diketahuinya nilai preferensi, dilakukan perangkingan dari tiap alternatif yang sudah dihitung. Jika nilai preferensi lebih besar dari yang lain, maka alternatif tersebut menjadi pilihan utama atau prioritas utama pada rekomendasi tersebut. Jika nilai preferensi tersebut dibawah dari lainnya, maka alternatif tersebut menjadi pilihan terakhir dari alternatif lainnya.

Tabel 4.17 Hasil Perangkingan

No	Alternatif Tempat Oleh-oleh	Preferensi	Ranking
1	Malang Strudel Sukarno Hatta 2	0,449691382	2
2	Lapis Malang Kawi	0,27568653	5

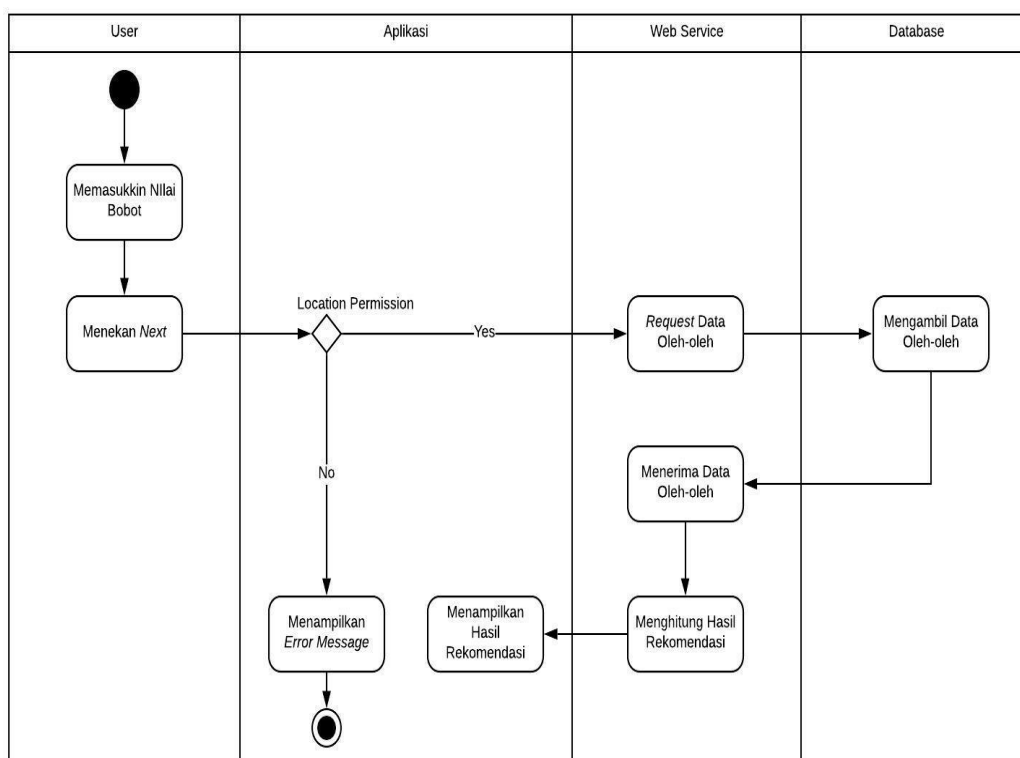


3	Pia Cap Mangkok	0,569800351	1
4	Pia Cap Mangkok Semeru	0,455317533	3
5	Goedang Oleh-oleh	0,391337112	4

Tabel 4.16 menunjukkan hasil dari preferensi dan juga perankingan sesuai dengan nilai yang paling tertinggi. Dimana didapatkan hasil rekomendasi dengan nilai tertinggi yaitu Pia Cap Mangkok dengan preferensi 0.569800351.

4.2.2 Perancangan Activity Diagram

Pada pengembangan yang dikerjakan ini, dibutuhkan perancangan tangan apa saja aktifitas yang sebelumnya dijelaskan pada Tabel 4.3 sampai Tabel 4.6 yang berisikan *use case scenario* dimana berisi perlakuan pengguna terhadap aplikasi yang dikembangkan..

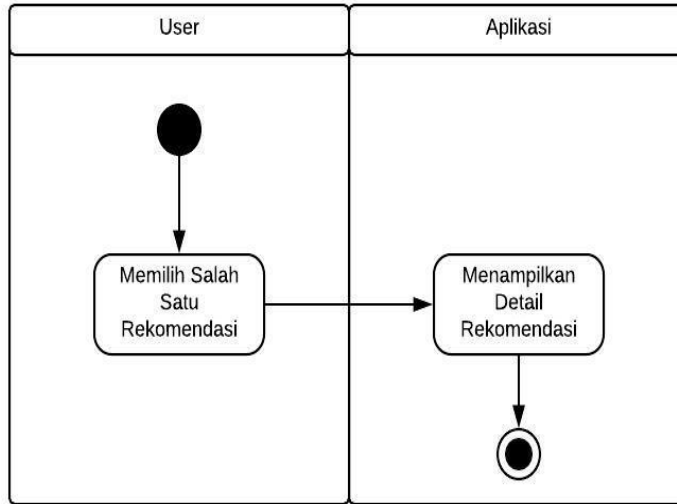


Gambar 4.6 Activity Diagram Memasukkan Bobot

Activity diagram Gambar 4.6 menunjukkan perlakuan user terhadap aplikasi begitu juga sebaliknya. User memasukkan nilai bobot dari kriteria yang ditampilkan pada aplikasi dengan nilai 1 – 10. Lalu user menekan next untuk melanjutkan aktifitas pada aplikasi. Jika user memperbolehkan akses lokasi/GPS, Aplikasi akan mengirim data bobot dan *Web Service* akan melakukan *request* data kepada Database untuk meminta data oleh – oleh. Database menerima *request*

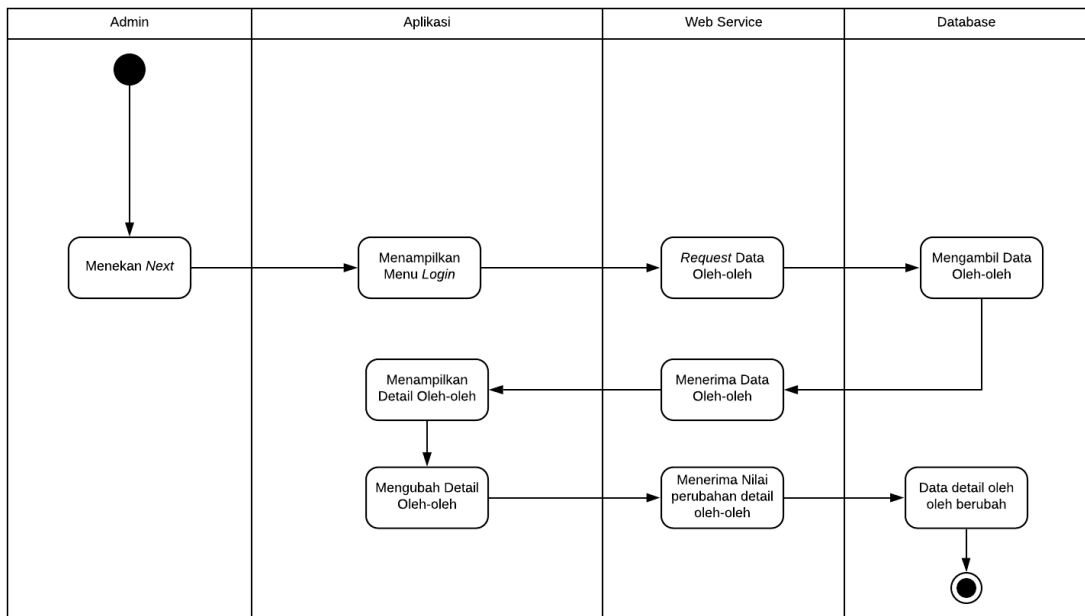


dari webservice, kemudia mengirim data oleh – oleh ke webservice yang selanjutkan akan ada perhitungan hasil rekomendasi dari bobot dan data yang telah diterima. Setelah dihitung, akan ditampilkan hasil rekomendasi ke aplikasi tersebut. Jika user tidak mengijinkan lokasi/GPS, maka akan tampil *error message*.



Gambar 4.7 Activity Diagram Menampilkan Hasil Rekomendasi

Activity diagram yang digambarkan pada Gambar 4.7 menyajikan aktifitas saat penampilan rekomendasi. Dimana saat user memilih rekomendasi dari hasil tersebut, maka aplikasi ditampilkanlah detail rekomendasi yang sesuai dengan pilihan user.



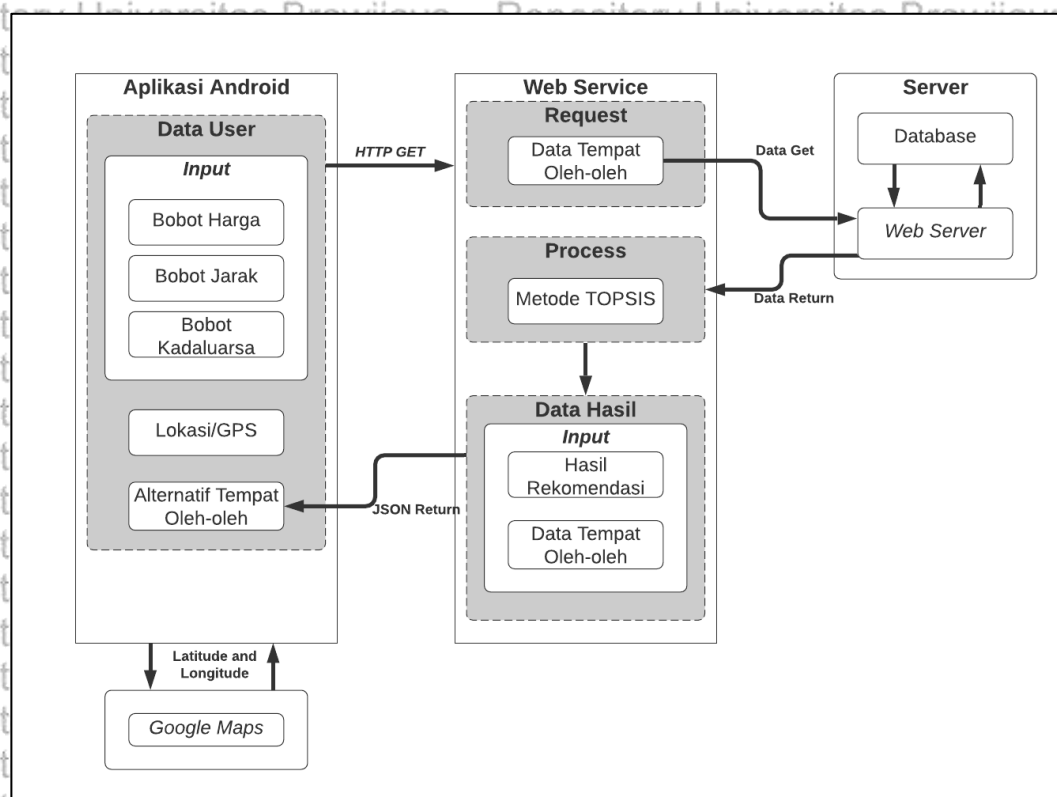
Gambar 4.8 Activity Diagram Mengubah Detail Tempat Oleh-Oleh



Activity diagram yang digambarkan pada Gambar 4.8 diatas merupakan aktifitas dimana admin melakukan perubahan atas detail oleh-oleh. Dimana admin pertama kali menekan tombol “admin” dimana akan dilanjutkan yaitu melakukan login sesuai dengan username dan password masing masing. Setelah itu data tempat oleh-oleh tersebut di-*request* agar dapat ditampilkan di aplikasi dan juga dapat diubah sesuai keinginan. Perubahan tersebut nantinya akan dikirimkan ke web service dan nilai pada database berubah sesuai dengan pemasukkan.

4.2.3 Perancangan Arsitektur Diagram

Salah satu perlakuan proses dalam perancangan perangkat lunak dimana diberikan penjelasan terhadap pembangunan sistem yang berada dalam masa *development* yaitu perancangan arsitektur pada sistem. Arsitektur sistem aplikasi rekomendasi oleh-oleh khas malang dijelaskan dengan gambar sebagai berikut.



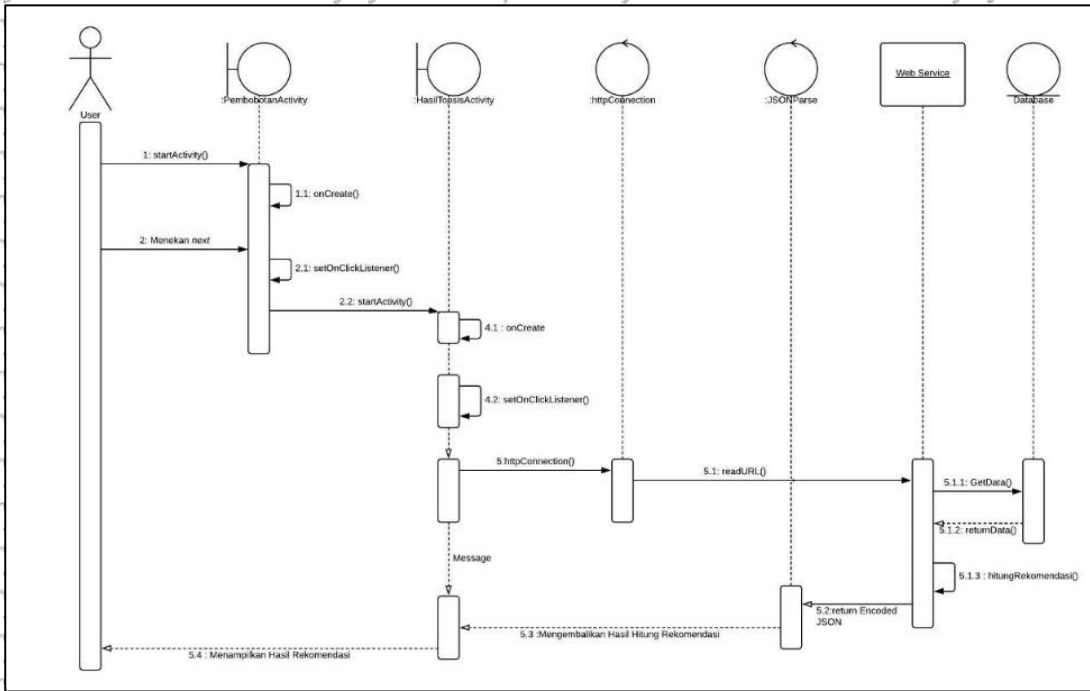
Gambar 4.9 Arsitektur Perangkat Bergerak

Perlakuan dari arsitektur perangkat bergerak Gambar 4.9 menampilkan aplikasi android sebagai media untuk user melakukan inputan dan juga tampilan hasil dari rekomendasi tersebut. Pada web service, data tempat oleh-oleh akan diminta ke database dimana akan diproses menggunakan metode topsis dan hasilnya merupakan JSON yang nantinya ditangkap oleh Aplikasi android. Penggunaan *Google Maps* yaitu untuk penyediaan data dari *latitude* dan *longitude* user untuk dikalkulasikan menggunakan TOPSIS yang berada pada *web service*.



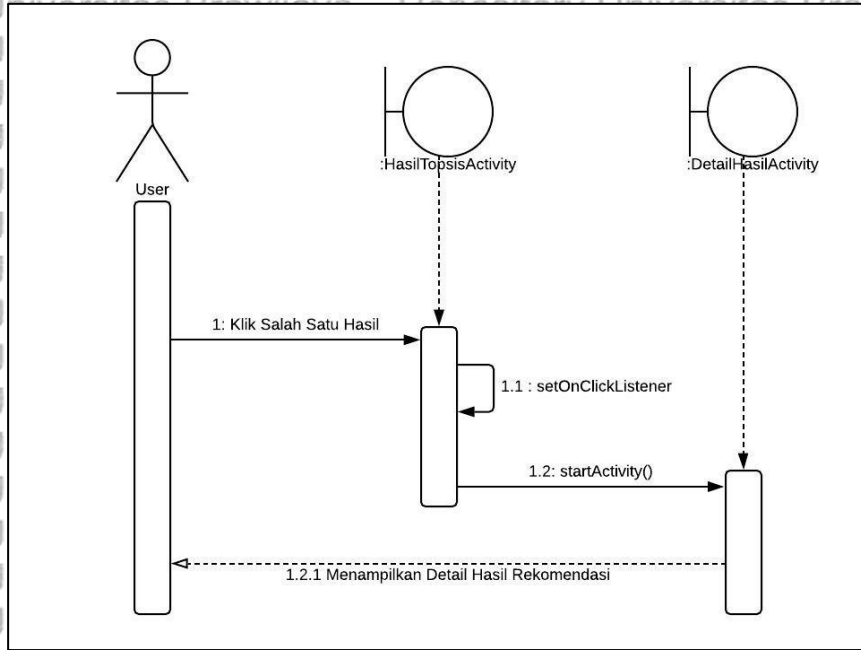
4.2.4 Perancangan Sequence Diagram

Sequence ini dirancang dan dijelaskan sebagaimana interaksi objek tersebut maupun juga kelas dimana terancang untuk aplikasi perangkat bergerak rekomendasi oleh-oleh yang dikembangkan.



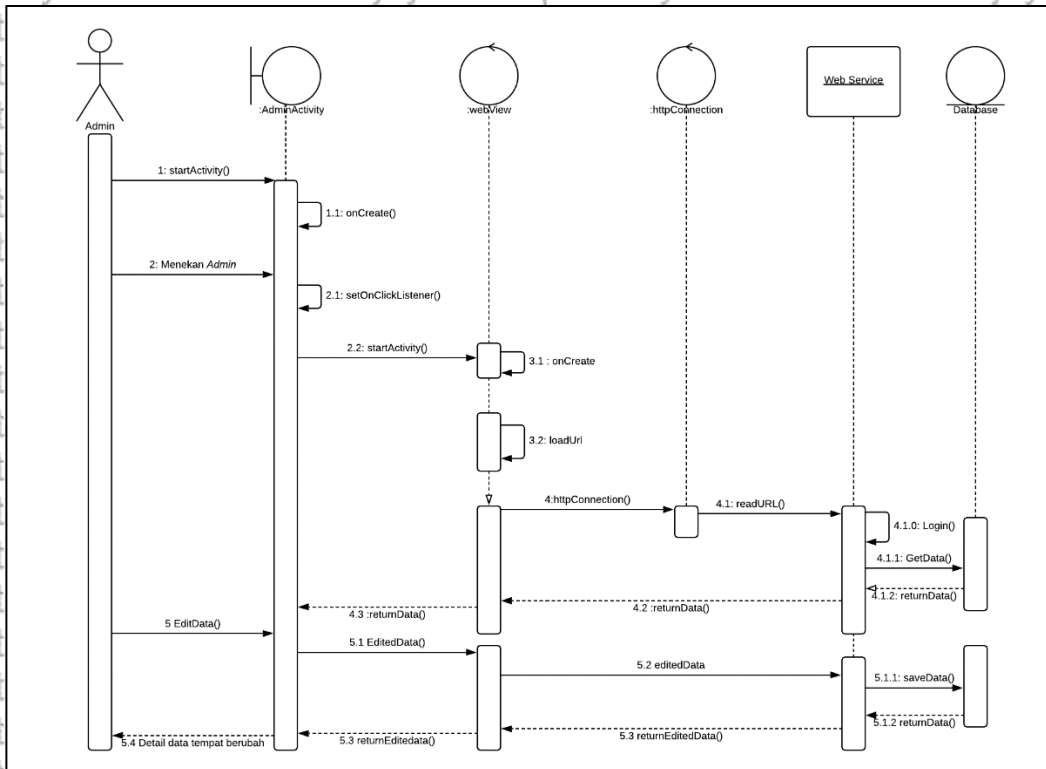
Gambar 4.10 *Sequence Diagram* Memasukkan Bobot dan Hasil Rekomendasi

Gambar 4.10 menampilkan *sequence diagram* memasukkan bobot kriteria sampai hasil keluaran rekomendasi. User pada awal membuka aplikasi, akan langsung menjalankan `startActivity` dan dilanjutkan oleh `onCreate`. Ketika user menekan `next`, maka akan menjalankan `setOnClickListener` yang akan memberikan data data dari inputan user sebelumnya. Inputan dari user akan diberikan ke `webservice` menggunakan `httpConnect`. `Web service` akan meminta data dari `database` yang nantinya akan dihitung dan dijadikan `JSON`. `JSON parser` akan mengembalikan data yang sudah dihitung ke aplikasi dan menampilkan hasil tersebut.



Gambar 4.11 Sequence Diagram Melihat Ringkasan

Gambar 4.11 menampilkan alur menampilkan detail rekomendasi dari hasil pada aktifitas sebelumnya.



Gambar 4.12 Sequence Diagram Mengubah Detail Tempat Oleh-oleh

Gambar 4.12 menampilkan sequence diagram mengubah detail tempat oleh-oleh dimana admin sebagai aktor yang melakukannya. Pertama admin masuk ke halaman admin activity lalu akan diberikan webview dimana halaman pertama

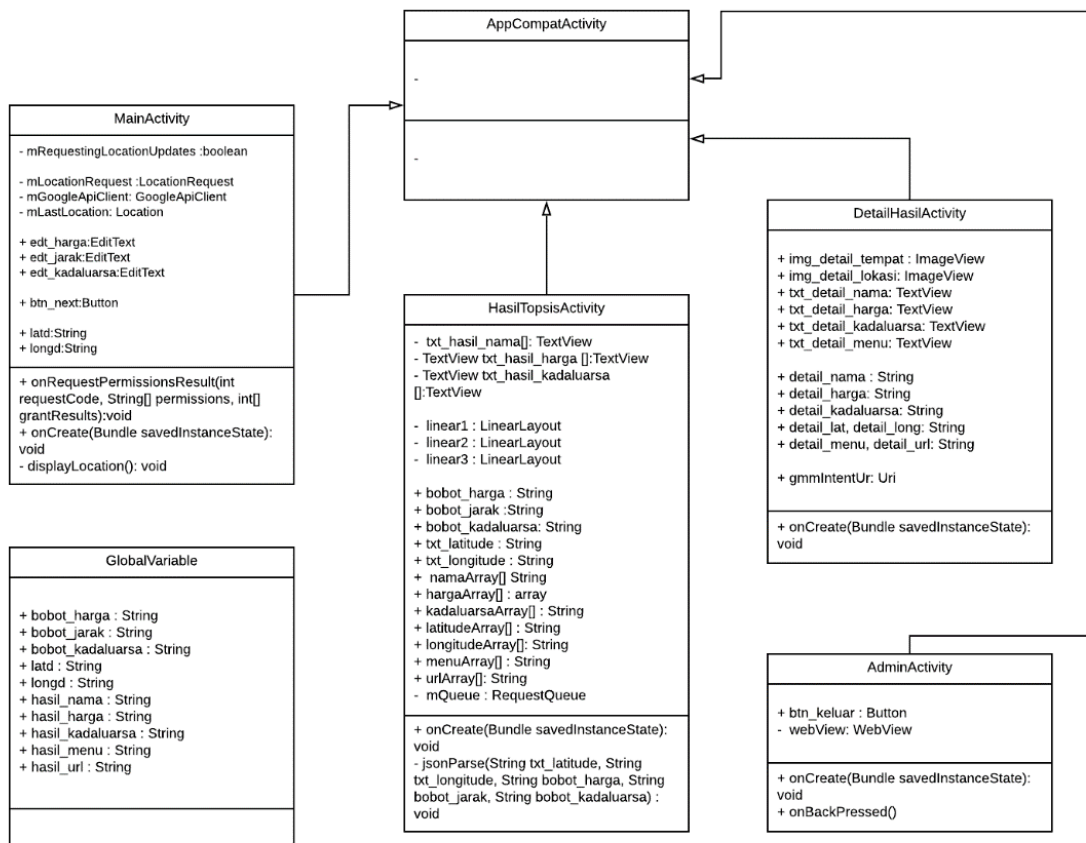


diberikan halaman login dimana terdapat *username* dan *password* sesuai dengan tempat oleh-oleh masing masing. Setelah itu ditampilkan halaman untuk melakukan perubahan dari detail tempat tersebut. Setelah data detail diambil melalui database, ditampilkan pada aplikasi dimana dapat dilakukan perubahan sesuai dengan keinginan admin. Jika sudah diubah, data tersebut dikembalikan ke database dan ditampilkan nilai sesuai dengan perubahan tersebut.

4.2.5 Perancangan *Class Diagram*

Didalam digram ini, ditampilkan data anggota penyusun penyusun yang akan dibangun pada suatu kelas di perangkat bergerak. Gambar dari perancangan kelas diagram terdapat di Gambar 4.13.

Gambar 4.13. menggambarkan perangkat bergerak sistem rekomendasi oleh-oleh yang berisikan kelas dari pengembangan tersebut. Kelas yang ada pada gambar tersebut adalah : *MainActivity*, *HasilTopsisActivity*, *DetailHasilActivity* dan *AdminActivity*. Dimana di kelas aktifitas tersebut terdapat variabel dan juga *method* yang digunakan. Ada juga *global variabel* yang digunakan untuk memberikan data ke aktivitas berikutnya.



Gambar 4.13 *Class Diagram* Perangkat Bergerak Sistem Rekomendasi Tempat Oleh-oleh



4.2.6 Perancangan Basis Data

Untuk mencapai tujuan pengembangan yang sesuai, dibutuhkan basis data untuk menyimpan data yang berisikan tempat oleh-oleh dimana telah disesuaikan sebagai pengembangan yang dikerjakan. Banyak tabel untuk aplikasi perangkat bergerak rekomendasi oleh-oleh terdapat satu table yang bernama tempat_oleh_oleh.

Tabel 4.18 Perancangan Basis Data tempat_oleh_oleh

tempat_oleh_oleh		
Nama	Tipe	Panjang
id	int	11
name	varchar	200
kondisi	varchar	200
latitude	double	200
longitude	double	200
harga	int	11
kadaluarsa	int	11
menu	varchar	200

Tabel 4.18 pada table diatas, terdapat 8 kolom yang terdiri dari: id sebagai *primary key*, name, kondisi, latitude, longitude, harga, kadaluarsa dan menu.

Tabel 4.19 Perancangan Basis Data admin_oleh_oleh

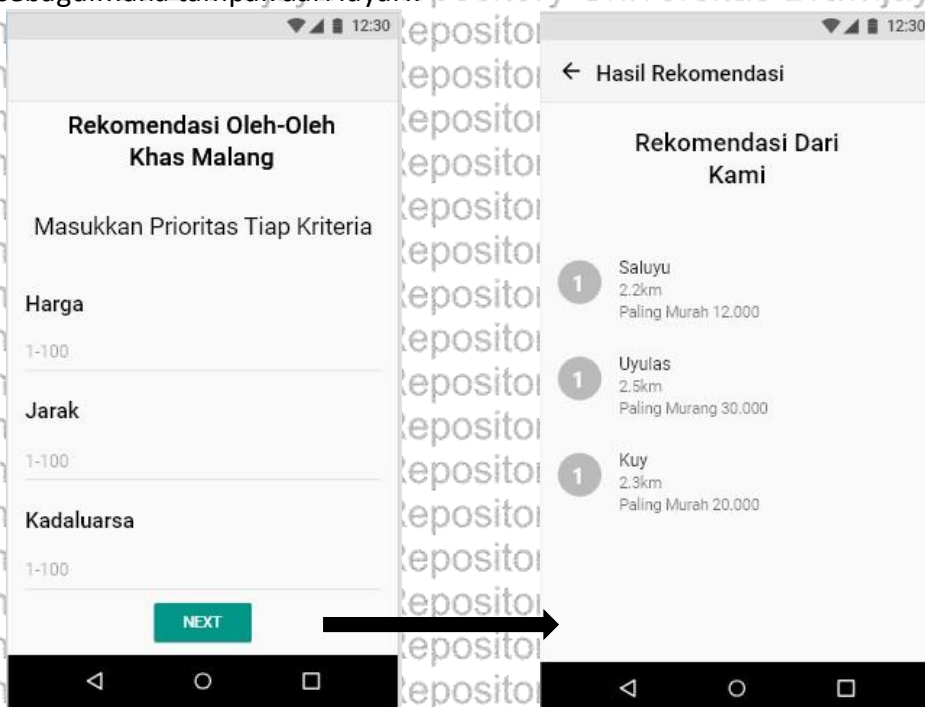
admin_oleh_oleh		
Nama	Tipe	Panjang
id	int	11
username	varchar	200
password	varchar	200
Tempat_oleh_oleh	double	200

Tabel 4.18 pada table diatas, terdapat 8 kolom yang terdiri dari: id sebagai *primary key*, username, password dan tempat_oleh_oleh sebagai indicator dari admin tersebut.

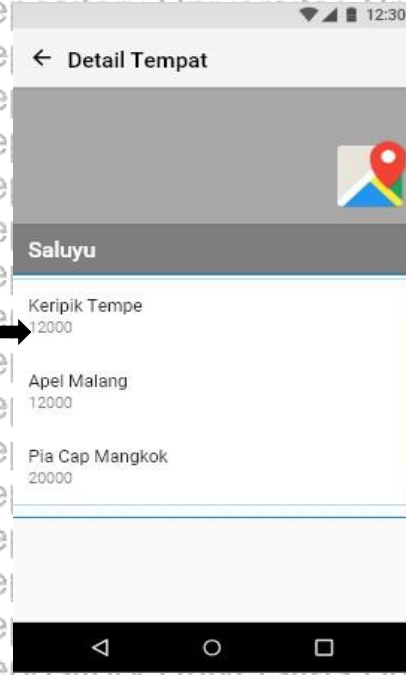


4.2.7 Perancangan *User Interface*

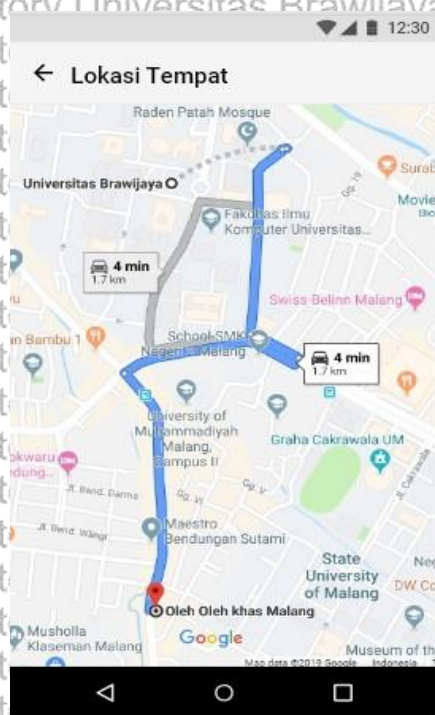
Setiap pengembangan aplikasi, antarmuka dibutuhkan untuk memperindah aplikasi yang dibuat dan menarik pengguna dalam tampilan yang ada. Begitu juga alur dari aplikasi tersebut dijelaskan sebagaimana aplikasi yang dirancang berjalan atau bisa disebut sebagai flow dari aplikasi tersebut. Dalam perancangan ini, diberikan gambaran dari aplikasi yang akan dibuat bersama dengan alur dari aplikasi sebagaimana tampak dari layar.



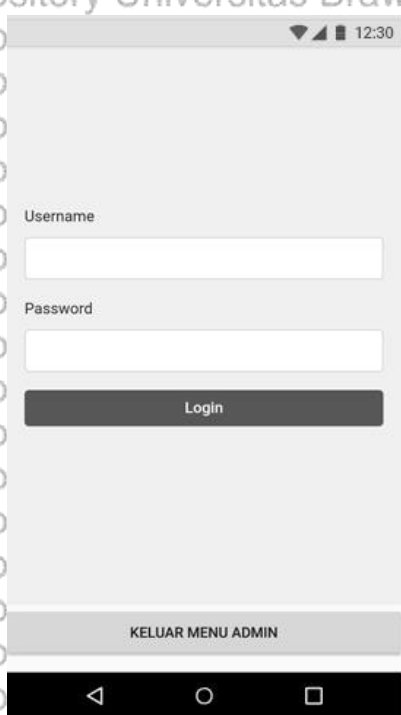
Gambar 4.14 Halaman Memasukkan Bobot dan Melihat Rekomendasi



Gambar 4.15 Halaman Melihat Rekomendasi dan Detail Tempat



Gambar 4.16 Halaman Melihat Detail tempat dan melihat lokasi tempat



Gambar 4.17 Halaman login dan Mengubah detail Oleh-oleh



a. Halaman Masukkan Bobot

Gambar 4.14 Bagian kiri adalah halaman pemasukkan bobot, dimana terdapat *input box* untuk pengguna memasukkan data bobot harga, jarak dan kadaluarsa. Tombol Next digunakan untuk melanjutkan aktifitas ke halaman berikutnya.

b. Halaman Melihat Rekomendasi

Gambar 4.15 Bagian kanan adalah halaman melihat rekomendasi, dimana ditampilkan 3 hasil rekomendasi dengan detail berupa jarak tempuh dari user dan harga oleh-oleh termurah dari tempat tersebut. Menekan salah satu list dari tempat rekomendasi tersebut untuk melanjutkan ke halaman selanjutnya.

c. Halaman Melihat Detail Tempat

Gambar 4.16 Bagian kanan adalah halaman melihat detail tempat, dengan detail yaitu foto dari tempat tersebut dan nama-nama oleh-oleh camilan beserta harga-harganya. Untuk melanjutkan ke halaman setelahnya, menekan ikon maps yang terdapat pada bagian samping kanan foto untuk melanjutkan ke halaman berikutnya.

d. Halaman Melihat Lokasi

Gambar 4.16 Bagian kanan adalah halaman untuk melihat lokasi tempat oleh-oleh dari halaman sebelumnya. Berisikan *maps* dan direksi pengguna sampai ke tempat tujuan.

e. Halaman Login

Gambar 4.17 Bagian kiri adalah halaman untuk melakukan login admin dengan username dan password yang berbeda beda.

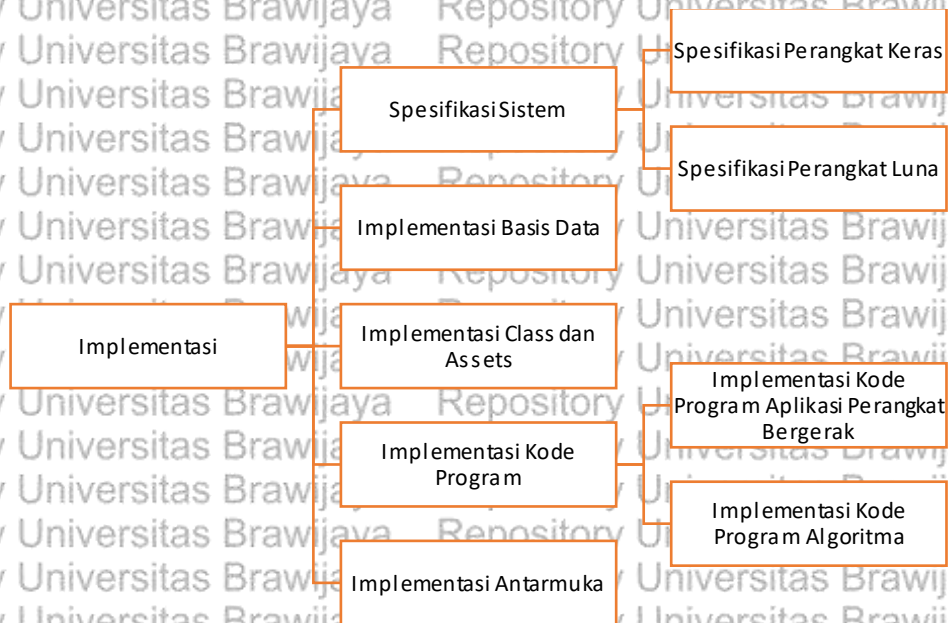
f. Halaman Mengubah Detail Oleh-Oleh

Gambar 4.17 Bagian kana adalah halaman untuk melakukan perubahan atas detail dari oleh-oleh di masing masing tempat.



BAB 5 IMPLEMENTASI

Didalam bab Implementasi, terdapat penjelasan dimana pengimplementasian dari apa yang telah dipaparkan sebelumnya yaitu analisis kebutuhan yang dimana telah di rancang juga pada bab silam. Langkah-langkah dan struktur dari implementasi akan di-ilustrasikan pada Gambar 5.1.



Gambar 5.1 Diagram Pohon Implementasi

Gambar diatas memberikan penjelasan salah satunya spesifikasi sistem dengan isian perangkat keras dan perangkat lunak yang dimana merupakan perangkat yang digunakan. Selain itu terdapat paparan tentang batasan berkenaan dengan pembuatan aplikasi, maupun pengimplementasian dari *use case* maupun basis data yang berisi kode pada perangkat bergerak dan algoritme TOPSIS dan terakhir adalah implementasi antarmuka atau interface.

5.1 Spesifikasi Sistem

Setiap mengembangkan sesuatu, dibutuhkan *device* yang dibutuhkan pada proses pengerjaan. Perangkat atau *device* tersebut mempunyai jenisnya masing masing yaitu perangkat keras dan perangkat lunak. Perangkat lunak berupa Sistem Operasi yang dimana dipunyai oleh masih masih perangkat keras dimana penelitian ini, perangkat keras berupa Laptop dan Perangkat Bergerak (*mobile device*).

5.1.1 Spesifikasi Perangkat Keras

Dijelaskan sebelumnya bahwa salah satu perangkat yang digunakan berupa laptop dengan spesifikasinya seperti berikut ini Tabel 5.1:



Tabel 5.1 Spesifikasi Perangkat Keras Laptop

Komponen	Spesifikasi
<i>System Model</i>	ASUS X450JN
<i>Processor</i>	Intel(R) Core(TM) i7-4710HQ @ CPU 2.50 GHz
<i>Memory</i>	8192MB DDR3
<i>Display</i>	NVIDIA GeForce GT 840M

Aplikasi pada pengujian di-instalasi dan dikerjakan pada *mobile device* dimana memiliki performansi dari system tersebut yaitu disebutkan di Tabel 5.2.

Tabel 5.2 Spesifikasi Mobile Device

Komponen	Spesifikasi
<i>System Model</i>	Xiaomi Redmi 4 Prime
<i>Processor</i>	Qualcom Snapdragon 625 @ 2.0 GHz
<i>Memory</i>	3072 MB
<i>Display</i>	Andreno 506

5.1.2 Spesifikasi Perangkat Lunak

Dijelaskan sebelumnya bahwa perangkat lunak adalah sistem operasi yang ada pada perangkat keras komputer/laptop yaitu dengan system operasi tersebut terdapat di Tabel 5.3.

Tabel 5.3 Spesifikasi Perangkat Lunak pada Laptop

Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Bahasa Pemrograman	Java Android dan PHP + HTML
<i>Tools Development</i>	Android Studio, Sublime Text, MySQL Workbench
<i>Tools Server dan DBMS</i>	XAMPP, MySQL
<i>Web Browser</i>	Google Chrome

Untuk system operasi yang ada pada *mobile device* yaitu sesuai dispesifikasikan pada Tabel 5.4.

Tabel 5.4 Spesifikasi Perangkat Lunak pada Mobile Device

Komponen	Spesifikasi
<i>Platform</i>	Android OS 4.0 atau lebih



5.2 Implementasi Basis Data

Pengimplementasian basis data pada pengembangan ini terpapar pada Gambar 4.18 dan 4.19 dimana menggunakan database management MySQL yang dikerjakan pada phpMyAdmin. Kode program untuk impelementasian basis data tersebut dilihat pada Kode 5.1 sampai dengan Kode 5.2.

Source Code Tabel basis data tempat oleh oleh

```
CREATE TABLE `topsis_oleh_oleh`.`tempat_oleh_oleh` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(200) NOT NULL ,
  `kondisi` VARCHAR(200) NOT NULL ,
  `longitude` DOUBLE NOT NULL ,
  `latitude` DOUBLE NOT NULL ,
  `harga` INT NOT NULL ,
  `kadaluarsa` INT NOT NULL ,
  `menu` VARCHAR(200) NOT NULL ,
  PRIMARY KEY (`id`))
```

Kode 5.1 Perintah SQL Membuat Tabel *topsis_oleh_oleh*

Kode 5.1 diatas adalah kode SQL dimana berisi perintah untuk membuat sebuah tabel dengan nama *tempat_oleh_oleh* dimana terdapat 1 *PRIMARY KEY* dan 5 kolom lainnya.

Source Code Tabel basis data admin oleh oleh

```
CREATE TABLE `topsis_oleh_oleh`.`admin_oleh_oleh` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `username` VARCHAR(200) NOT NULL ,
  `password` VARCHAR(200) NOT NULL ,
  `tempat_oleh_oleh` VARCHAR(200) NOT NULL ,
  PRIMARY KEY (`id`))
```

Kode 5.2 Perintah SQL Membuat Tabel *admin_oleh_oleh*

Kode 5.1 diatas adalah kode SQL dimana berisi perintah untuk membuat sebuah tabel dengan nama *tempat_oleh_oleh* dimana terdapat 1 *PRIMARY KEY* dan 5 kolom lainnya.

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

Implemenasi dari kelas dengan aset dari rancangan telah dijelaskan pada diagram kelas pada gambar 4.13 . Untuk implementasi dari rancangan ini yaitu



Android terdapat file-file yang berbeda untuk tiap pengimplementasian dimana Java sebagai bagian untuk melakukan operasi logika dan xml untuk tampilan layout pada aplikasi. Lebih lanjutnya dapat dipaparkan di Tabel 5.5

Tabel 5.5 Berkas Kelas dan Assets Layout pada Aplikasi Perangkat Bergerak

No.	Package	Folder	Nama Kelas	Nama Layout
1.	com.example.rekomendasiolaholeh		MainActivity.java	activity_main.xml
2	com.example.rekomendasiolaholeh		HasilTopsisActivity.java	activity_hasil_topsis.xml
3	com.example.rekomendasiolaholeh		DetailHasilActivity.java	activity_detail_hasil.xml
4	com.example.rekomendasiolaholeh		AdminActivity.java	activity_admin.xml

5.4 Implementasi Kode Program

Bagian-bagian dari Tabel 5.5 akan dijelaskan bagaimana kode tersebut diimplementasikan dengan tambahan web service dan algoritme TOPSIS. Rincian lebih lanjut dapat dilihat berikut.

5.4.1 Implementasi Kode Program Aplikasi Perangkat Bergerak

Program yang dilakukan pengembangannya ini berasal dari fungsi-fungsi yang telah di analisis dan dirancang pada bab silam. Dari fungsionalitas yang telah direncanakan sebelumnya, pada package app.example.rekomendasiolaholeh dirancang sebagaimana menyesuaikan dari class diagram pada Gambar 4.13.

No	MainActivity
1	public void onRequestPermissionsResult (int requestCode, String[]
2	permissions, int[] grantResults) {
3	switch (requestCode) {
4	case MY_PERMISSION_REQUEST_CODE:
5	if (grantResults.length > 0 && grantResults[0] ==
6	PackageManager.PERMISSION_GRANTED) {
7	if (checkPlayServices ())
8	buildGoogleApiClient ();
9	}
10	break;
11	}
12	}
13	
14	if (ActivityCompat.checkSelfPermission (this,
15	Manifest.permission.ACCESS_FINE_LOCATION) !=
16	PackageManager.PERMISSION_GRANTED
17	&& ActivityCompat.checkSelfPermission (this,
18	Manifest.permission.ACCESS_COARSE_LOCATION) !=
19	PackageManager.PERMISSION_GRANTED) {
20	
21	ActivityCompat.requestPermissions (this, new String[] {
22	Manifest.permission.ACCESS_FINE_LOCATION,
23	Manifest.permission.ACCESS_COARSE_LOCATION,
24	}, MY_PERMISSION_REQUEST_CODE);
25	} else {



```

26         if (checkPlayServices()) {
27             buildGoogleApiClient();
28             createLocationRequest();
29         }
30     }
31 }
32 displayLocation();
33
34
35     private void displayLocation() {
36         if (ActivityCompat.checkSelfPermission(this,
37 Manifest.permission.ACCESS_FINE_LOCATION) !=
38 PackageManager.PERMISSION_GRANTED
39         && ActivityCompat.checkSelfPermission(this,
40 Manifest.permission.ACCESS_COARSE_LOCATION) !=
41 PackageManager.PERMISSION_GRANTED) {
42             return;
43         }
44         mLokasiHPTerakhir =
45 LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient
46 );
47         if (mLokasiHPTerakhir != null) {
48             double lati = mLokasiHPTerakhir.getLatitude();
49             double longi = mLokasiHPTerakhir.getLongitude();
50             latiDude = String.valueOf(lati);
51             longiDude = String.valueOf(longi);
52         } else {
53
54         }
55     }

```

Kode 5.3 Kode Program Kelas *MainActivity*

1. Baris 1 sampai 11, untuk melakukan *request* permission ke device pengguna.
2. Baris 13-28, untuk melakukan akses dari *request* GPS pada device.
3. Baris 33-51, untuk melakukan mengambil data *latitude* dan *longitude* pengguna dari permission yang sesuai.

No	HasilTopsisActivity
1	private void jsonParse(String txt_latitude, String
2	txt_longitude, String bobot_oleholeh_harga, String
3	bobot_oleholeh_jarak, String bobot_oleholeh_kadaluarsa){
4	
5	String url =
6	"http://topsisfhj.xyz/TOPSIS_OLEH_OLEH/test_json_real_device.php?l
7	at="+txt_latitude+"&long="+txt_longitude+"&harga="+bobot_oleholeh_
8	harga_oleholeh+"&jarak="+bobot_oleholeh_jarak+"&kadaluarsa="+bobot
9	_oleholeh_kadaluarsa;
10	
11	JSONArrayRequest request = new
12	JSONArrayRequest(Request.Method.GET, url, null,
13	new Response.Listener<JSONArray>() {
14	@Override
15	public void onResponse(JSONArray response) {
16	try {
17	
18	for (int i = 0; i < response.length();
19	i++) {
20	
21	JSONObject jsonTest =
22	response.getJSONObject(i);
23	



```

24         namaArray[i]
25         jsonTest.getString("name");
26         hargaArray[i]
27         jsonTest.getString("harga");
28         kadaluarsaArray[i]
29         jsonTest.getString("kadaluarsa");
30         latitudeArray[i]
31         jsonTest.getString("latitude");
32         longitudeArray[i]
33         jsonTest.getString("longitude");
34         menuArray[i]
35         jsonTest.getString("menu");
36         urlArray[i]
37         jsonTest.getString("url");
38
39
40         txt_hasil_nama[i].setText(namaArray[i]);
41
42         txt_hasil_harga[i].setText(hargaArray[i]);
43
44         txt_hasil_kadaluarsa[i].setText(kadaluarsaArray[i]);
45
46     }
47
48     } catch (JSONException e) {
49         e.printStackTrace();
50     }
51 }
52 }, new Response.ErrorListener() {
53     @Override
54     public void onErrorResponse(VolleyError error) {
55         error.printStackTrace();
56     }
57 }
58 });
59
60     mQueue.add(request);
61 }

```

Kode 5.4 Kode Program Kelas *HasilTopsisActivity*

1. Baris 1 sampai 59, untuk melakukan akses ke JSON dari url yang sudah ada dan mengembalikan nilai string untuk tiap datanya.

No	DetailHasilActivity
1	Picasso.get().load(detail_url).into(img_detail_tempat);
2	
3	img_detail_lokasi.setOnClickListener(new
4	View.OnClickListener() {
5	@Override
6	public void onClick(View view) {
7	
8	gmmIntentUri
9	Uri.parse("google.navigation:q="+detail_lat+","+detail_long);
10	Intent mapIntent = new Intent(Intent.ACTION_VIEW,
11	gmmIntentUri);
12	mapIntent.setPackage("com.google.android.apps.maps");
13	
14	



15	if (mapIntent.resolveActivity(getPackageManager().
16	!= null) {
17	startActivity(mapIntent);
18	}
19	};
20	});
21	}
22	}

Kode 5.5 Kode Program Kelas *DetailHasilActivity*

1. Baris 1, untuk menampilkan gambar dari url yang ada.
2. Baris 3 – 16, untuk mengarahkan ke aplikasi Maps yang langsung ditunjukkan lokasi yang dituju.

No	AdminActivity
1	webView = (WebView) findViewById(R.id.webviewAdmin);
2	webView.setWebViewClient(new WebViewClient());
3	
4	webView.loadUrl("http://topsisfhj.xyz/TOP SIS_OLEH_OLEH/admin/login
5	.php");
6	
7	WebSettings webSettings = webView.getSettings();
8	webSettings.setJavaScriptEnabled(true);
9	
10	btn_keluar.setOnClickListener(new View.OnClickListener() {
11	@Override
12	public void onClick(View v) {
13	Intent intentToMain = new Intent(v.getContext(),
14	MainActivity.class);
15	startActivity(intentToMain);
16	}
17	});

Kode 5.6 Kode Program Kelas *AdminActivity*

1. Baris 1 - 8, untuk menampilkan webview sesuai dengan url yang dituju.

5.4.2 Implementasi Kode Program Algoritme TOPSIS

Pengimplementasian metode ini menggunakan PHP dimana dikerahkan saat aplikasi dari android memanggil php tersebut dimana sesuai dengan Gambar 4.3. Kode pemrograman Metode tersebut sesuai dengan dibawah ini.

No	Menghitung pembagi
1	function getJarakAntar(\$latit1, \$longit1, \$latit2,
2	\$longit2){
3	\$earthRadius=6371000;
4	\$radLatit1 = deg2rad(\$latit1);
5	\$radLongit1 = deg2rad(\$longit1);
6	\$radLatit2 = deg2rad(\$latit2);
7	\$radLongit2 = deg2rad(\$longit2);
8	
9	\$deltaLatit = \$radLatit2 - \$radLatit1;
10	\$deltaLongit = \$radLongit2 - \$radLongit1;



```

11     $c = 2 * asin(sqrt(pow(sin($deltaLatit / 2), 2) +
12     cos($radLatit1) * cos($radLatit2) * pow(sin($deltaLongit
13     / 2), 2)));
    return $c * $earthRadius;
}
    
```

Kode 5.7 Kode Program Menghitung nilai Haversine

Kode 5.7 adalah kode pencarian jarak antara dua latitude dengan longitude yang berbeda. Untuk penjelasan dari implementasi tersebut berisikan:

1. Baris 1 – 13, untuk menghitung nilai haversine dari dua latitude dan longitude yang berbeda.

No	Menghitung pembagi
1	for(\$t = 0; \$t < count(\$result); \$t++){
2	
3	\$temptBagian[0] += pow(\$result[\$t]["harga"], 2);
4	\$temptBagian[1] +=
5	pow(getJarakAntar(\$result[\$t]["latitude"], \$result[\$t]["long
6	itude"), \$lat, \$long), 2);
7	\$temptBagian[2] += pow(\$result[\$t]["kadaluarsa"], 2);
8	
9	}
10	\$division =
	array(sqrt(\$temptBagian[0]), sqrt(\$temptBagian[1]), sqrt(\$tem
	ptBagian[2]));

Kode 5.8 Kode Program Menghitung nilai pembagi

Kode 5.8 menunjukkan implementasian pencarian nilai pembagi. Untuk penjelasan dari implementasi kode tersebut yaitu:

2. Baris 1 sampai dengan 9 untuk melakukan looping dari banyaknya alternatif yang ada dan untuk menjumlahkan pengkuadratan dari kriteria yang ada.
3. Baris 10 untuk menyimpan data array dari tiga pembagi dengan isian pengakarkan dari hasil jumlah kuadrat pada looping sebelumnya.

No	Menghitung normalisasi
1	for(\$t = 0; \$t < count(\$result); \$t++){
2	\$HargaNormalis[\$t] = \$result[\$t]["harga"] / \$division[0];
	\$JarakNormalis[\$t] =
3	getJarakAntar(\$result[\$t]["latitude"], \$result[\$t]["longitud
	e"), \$lat, \$long) / \$division[1];
	\$KadaluarsaNormalis[\$t] = \$result[\$t]["kadaluarsa"]
4	/ \$division[2];
5	}

Kode 5.9 Kode Program Menghitung Nilai Normalisasi

Kode 5.9 adalah untuk mencari nilai normalisasi dimana akan dijelaskan kode tersebut dibawah ini:

1. Baris 1-5, untuk melakukan looping dengan banyaknya alternatif(array result) yang ada dengan nilai tiap kriteria (harga, jarak, kadaluarsa) dibagi dengan nilai dari pembagi yang menghasilkan normalisasi.



No	Menghitung pembobot
1	for(\$t = 0; \$t < count(\$HargaNormalis); \$t++){
2	\$HargaHitPembobot[\$t] = \$HargaNormalis[\$t] *
3	\$tnputBobotHarga; \$JarakHitPembobot[\$t] = \$JarakNormalis[\$t] *
4	\$tnputBobotJarak; \$KadaluarsaHitPembobot[\$t] = \$KadaluarsaNormalis[\$t] *
5	\$tnputBobotKadaluarsa; }

Kode 5.10 Kode Program Menghitung Nilai Pembobot

Kode 5.10 adalah pencarian nilai pembobot, dimana akan dijelaskan implementasian tersbut dibawah ini:

1. Baris 1 sampai 5 untuk melakukan looping dengan banyaknya normalisasi yang ada dengan tiap nilai terbobot pada tiap kriteria berisi nilai dari normalisasi tiap bobot dengan inputan bobot.

No	Menghitung a+ dan a- dengan si+ dan si-
1	\$plusHargaHitPembobot = Min(\$HargaHitPembobot);
2	\$minusHargaHitPembobot = Max(\$HargaHitPembobot);
3	
4	\$plusJarakHitPembobot = Min(\$JarakHitPembobot);
5	\$minusJarakHitPembobot = Max(\$JarakHitPembobot);
6	
7	\$plusKadaluarsaHitPembobot = Max(\$KadaluarsaHitPembobot);
8	\$minusKadaluarsaHitPembobot = Min(\$KadaluarsaHitPembobot);
9	
10
11	
12	for(\$t = 0; \$t < count(\$HargaHitPembobot); \$t++){
13	\$sPlus[\$t] = sqrt(pow(\$HargaHitPembobot[\$t] - \$plusHargaHitPembobot, 2) + pow(\$JarakHitPembobot[\$t] - \$plusJarakHitPembobot, 2) + pow(\$KadaluarsaHitPembobot[\$t] - \$plusKadaluarsaHitPembobot, 2));
14	\$sMin[\$t] = sqrt(pow(\$HargaHitPembobot[\$t] - \$minusHargaHitPembobot, 2) + pow(\$JarakHitPembobot[\$t] - \$minusJarakHitPembobot, 2) + pow(\$KadaluarsaHitPembobot[\$t] - \$minusKadaluarsaHitPembobot, 2));
	}

Kode 5.11 Kode Program Menghitung Nilai a+ dan a- dengan si+ dan si-

Kode 5.11 untuk Pencarian nilai a+ dan a- dengan si+ dan si-, dimana akan dijelaskan implementasian tersbut dibawah ini:

1. Baris 1-8 merupkan penghitungan nilai a+ dan a- menggunakan max untuk a+ dan min untuk a-
2. Baris 12-14 untuk menghitung nilai s+ dan s- dengan melakukan looping banyaknya array HargaHitPembobot berisi pengakaran dari jumlah kuadrat tiap bobot dikurangan nilai a+ untuk s+ dan dikurangan a- untuk s-



No	Menghitung jumlah SI dan ci+
1	<code>\$jumlahSI = array();</code>
2	<code>for(\$t=0; \$t < count(\$sPlus); \$t++){</code>
3	<code> \$jumlahSI[\$t] = \$sPlus[\$t] + \$sMin[\$t];</code>
4	<code>}</code>
5	<code>\$valCi = array();</code>
6	<code>for(\$t = 0; \$t < count(\$jumlahSI); \$t++){</code>
7	<code> \$valCi[\$t] = \$sMin[\$t] / \$jumlahSI[\$t];</code>
7	<code>}</code>

Kode 5.12 Kode Program Menghitung jumlah SI dan nilai ci+

Kode 5.12 untuk mencari jumlah nilai SI dan mencari nilai ci+, dimana akan dijelaskan implementasian tersbut dibawah ini:

1. Baris 1-3 untuk menghitung jumlah SI dengan looping banyaknya data s+ berisi penjumlahan s+ dengan s-
2. Baris 5-7 untuk menghitung jumlah ci dengan looping banyaknya data jumlah SI yang berisi s- dibagi dengan nilai dari jumlah SI.

No	Mengurutkan nilai ci+
1	<code>for (\$t=count(\$valCi); \$t >=0 ; \$t--) {</code>
2	<code> for (\$y=1; \$y < \$t; \$y++){</code>
3	<code> if (\$valCi[\$y-1]<\$valCi[\$y]) {</code>
4	<code> \$temp=\$valCi[\$y-1];</code>
5	<code> \$valCi[\$y-1]=\$valCi[\$y];</code>
6	<code> \$valCi[\$y]=\$temp;</code>
7	<code> \$temp2=\$result[\$y-1];</code>
8	<code> \$result[\$y-1]=\$result[\$y];</code>
9	<code> \$result[\$y]=\$temp2;</code>
10	<code> }</code>
11	<code> }</code>
12	<code>}</code>
13	
14	<code>\$finalResult=array();</code>
15	<code>for (\$t=0; \$t < \$jumlahRekomendasi; \$t++){</code>
16	<code> \$finalResult[]=\$result[\$t];</code>
17	<code>}</code>

Kode 5.13 Kode Program Mengurutkan Nilai dari ci+

Kode 5.13 untuk mengurutkan nilai ci+. Penjelasan implementasi kode program sebagai berikut:

1. Baris 1-12 untuk melakukan pengurutan nilai ci+ dari data yang telah dihitung sebelumnya.
2. Baris 14-17 untuk melakukan pemasukkan data yang telah diurutkan dengan banyaknya jumlah rekomendasi yang diinginkan.



5.5 Implementasi Antarmuka

Antarmuka yang sebelum telah dirancang menyesuaikan perancangan Gambar 4.14 sampai 4.17 dilakukan implementasi yang menyesuaikan. Pada implementasi ini, ditunjukkan hasil secara tampak layar bagaimana aplikasi tersebut dalam kondisi yang real. Gambar 5.2 sampai 5.5 menjelaskan bagaimana tiap halaman berinteraksi satu sama lain.

12:47

Rekomendasi Oleh Oleh

Rekomendasi Oleh Oleh Khas Malang

Masukkan Nilai Sesuai Prioritas Anda

Harga	-	5	+
Mengutamakan Harga Termurah			
Jarak	-	3	+
Mengutamakan Jarak Terdekat			
Kadaluarsa	-	2	+
Mengutamakan Kadaluarsa Terlama			

Semakin besar nilai prioritas, Semakin diutamakan

NEXT

ADMIN

Gambar 5.2 Halaman Memasukkan Bobot

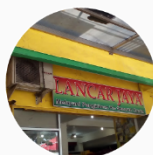
Gambar 5.2 adalah halaman pemasukkan bobot, dimana terdapat *input box* untuk pengguna memasukkan data bobot harga, jarak dan kadaluarsa. Tombol Next digunakan untuk melanjutkan aktifitas ke halaman berikutnya. Untuk memasuki halaman admin, dapat menekan tombol admin dibawah tombol nex.



12:51

← Rekomendasi Oleh Oleh

Rekomendasi Dari Kami



Lancar Jaya Tempe and Fruit Chips

Paling Murah : Rp.15000,-
Kadaluarsa Paling Cepat : 20 Hari



Pia Cap Mangkok Soehat (TUTUP)

Paling Murah : Rp.11500,-
Kadaluarsa Paling Cepat : 12 Hari

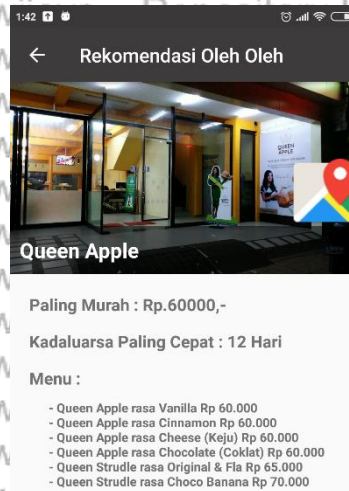


Kripik Tempe Rizky Barokah

Paling Murah : Rp.5000,-
Kadaluarsa Paling Cepat : 11 Hari

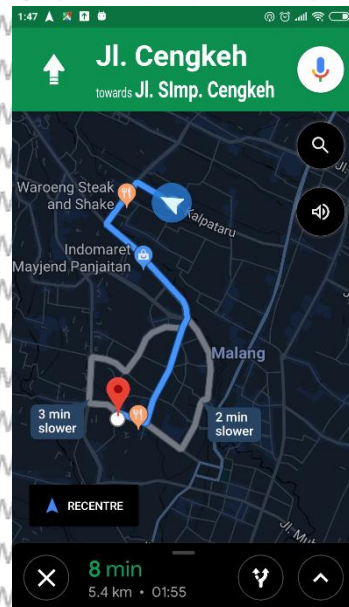
Gambar 5.3 Halaman Melihat Rekomendasi

Gambar 5.3 adalah halaman melihat rekomendasi dimana ditampilkan 3 hasil rekomendasi dengan detail berupa jarak tempuh dari user dan harga oleh-oleh termurah dari tempat tersebut. Menekan salah satu list dari tempat rekomendasi tersebut untuk melanjutkan ke halaman selanjutnya.



Gambar 5.4 Halaman Melihat Detali tempat

Gambar 5.4 adalah halaman melihat detail tempat, dengan detail yaitu foto dari tempat tersebut dan nama-nama oleh-oleh camilan beserta harga-harganya. Untuk melanjutkan ke halaman setelahnya, menekan ikon maps yang terdapat pada bagian samping kanan foto untuk melanjutkan ke halaman berikutnya.



Gambar 5.5 Halaman Melihat Detali tempat

Gambar 5.5 adalah halaman untuk melihat lokasi tempat oleh-oleh dari halaman sebelumnya. Berisikan *maps* dan direksi pengguna sampai ke tempat tujuan.



Gambar 5.6 Halaman Login

Gambar 5.6 adalah halaman login untuk admin jika ingin melakukan perubahan pada tempat oleh-oleh.

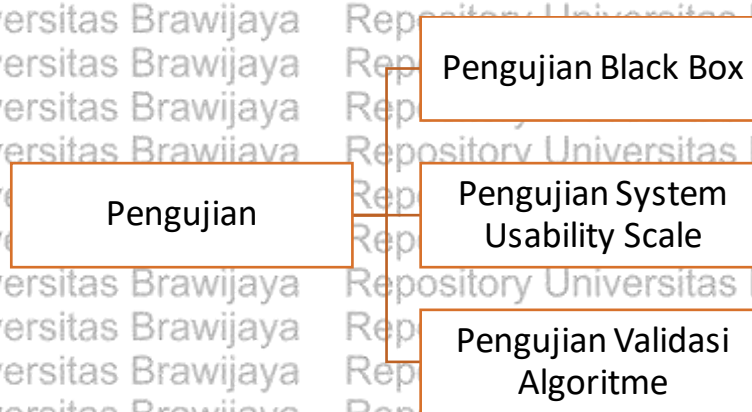
Gambar 5.5 Halaman Mengubah Detail Tempat Oleh-oleh

Gambar 5.5 adalah halaman untuk mengubah detail dari tempat oleh-oleh sesuai dengan admin tersebut.



BAB 6 PENGUJIAN DAN ANALISIS

Ketika aplikasi sudah dilakukan implementasi, aplikasi yang sudah dikembangkan tersebut harus diuji mulai dari fungsionalitas sampai nonfungsionalitas. Gambar 6.1 menggambarkan *testing* apa saja yang akan dilakukan pada aplikasi perangkat bergerak yang diimplementasikan yaitu rekomendasi oleh-oleh menggunakan metode TOPSIS.



Gambar 6.1 Pohon Diagram Pengujian dan Analisis

Testing yang dikerahkan pada aplikasi yang dikembangkan yaitu: *black box test* yaitu pengujian fungsionalitas dari sistem, *SUS test* dan pengujian algoritme untuk melakukan pengecekan valid atau tidaknya algoritme tersebut.

6.1 Pengujian *Black-Box*


Pengujian dimana teknik untuk melakukannya berasal dari fungsionalitas dimana merujuk kepada *use case scenario* sebagai alat bantu untuk melakukan pengujian. Dengan adanya pengujian ini, akan terpaparkan apakah aplikasi yang dikembangkan tersebut sudah memenuhi kriteria dari kebutuhan yang ada atau belum. Pengujian yang sudah sesuai dengan kebutuhan bernilai valid dan bernilai tidak valid jika kebutuhan tersebut tidak terpenuhi.

Tabel 6.1 sampai 6.4 merupakan pemberian kejelasan yang berisi hasil fungsional *black box* yang diujikan.

Tabel 6.1 Pengujian Fungsional Memasukkan Bobot Kriteria

Nomor Kasus Uji	TOB-01
Nama Kasus Uji	Memasukkan Bobot Kriteria
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memberikan kapasitas pada pemasukkan bobot kriteria bekerja
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka aplikasi rekomendasi oleh-oleh. 2. Berada pada halaman utama dari rekomendasi oleh-oleh.



	<p>3. Bobot yang tersedia dimasukkan sesuai kriteria pengguna.</p> <p>4. Menekan tombol dibawah berupa <i>next</i> untuk melanjutkan kehalaman berikutnya</p>
<p>Hasil yang Diharapkan</p>	<p>Pengguna memasukkan bobot kriteria yang diinginkanh oleh pengguna.</p>
<p>Screenshot Main Flow</p>	

Gambar 6.2 Hasil Pengujian Masukkan bobot

Tabel 6.1 menjelaskan mengenai prosedur melakukan pemasukkan bobot kriteria.

Tabel 6.2 Pengujian Fungsional Melihat Hasil TOPSIS

<p>Nomor Kasus Uji</p>	<p>TOB-02</p>
<p>Nama Kasus Uji</p>	<p>Melihat Hasil Rekomendasi Tempat Oleh - Oleh</p>
<p>Objek Uji</p>	<p>Kebutuhan Fungsional</p>
<p>Tujuan Pengujian</p>	<p>Untuk memberika kepastian pada aplikasi yang dikembangkan menampilkan rekomendasi dari hasil metode tersebut.</p>
<p>Prosedur Pengujian</p>	<p>1. Membuka aplikasi rekomendasi oleh-oleh.</p>



	<ol style="list-style-type: none"> 2. Berada pada halaman utama dari rekomendasi oleh-oleh. 3. Bobot yang tersedia dimasukkan sesuai kriteria pengguna. 4. Menekan tombol dibawah berupa next untuk melanjutkan kehalaman berikutnya 5. Mendapatkan beberapa rekomendasi oleh-oleh yang disediakan.
<p>Hasil yang Diharapkan</p>	<p>Pengguna melihat beberapa rekomendasi dari hasil TOPSIS yang dilakukan.</p>
<p>Screenshot</p>	 <p>The screenshot shows a mobile application interface titled 'Rekomendasi Oleh Oleh'. It features three recommended items, each with a circular image, a title, and pricing information:</p> <ul style="list-style-type: none"> Lancar Jaya Tempe and Fruit Chips: Paling Murah : Rp.15000,-, Kadalursa Paling Cepat : 20 Hari Pia Cap Mangkok Soehat (TUTUP): Paling Murah : Rp.11500,-, Kadalursa Paling Cepat : 12 Hari Kripik Tempe Rizky Barokah: Paling Murah : Rp.5000,-, Kadalursa Paling Cepat : 11 Hari

Gambar 6.3 Hasil Pengujian Melihat hasil TOPSIS

Tabel 6.3 adalah penjelasan bagaimana fungsionalitas dari melihat Hasil Rekomendasi diuji.

Tabel 6.3 Pengujian Fungsional Melihat Detail Rekomendasi

<p>Nomor Kasus Uji</p>	<p>TOB-03</p>
<p>Nama Kasus Uji</p>	<p>Melihat Detail Hasil Rekomendasi Tempat Oleh-Oleh</p>
<p>Objek Uji</p>	<p>Kebutuhan Fungsional</p>



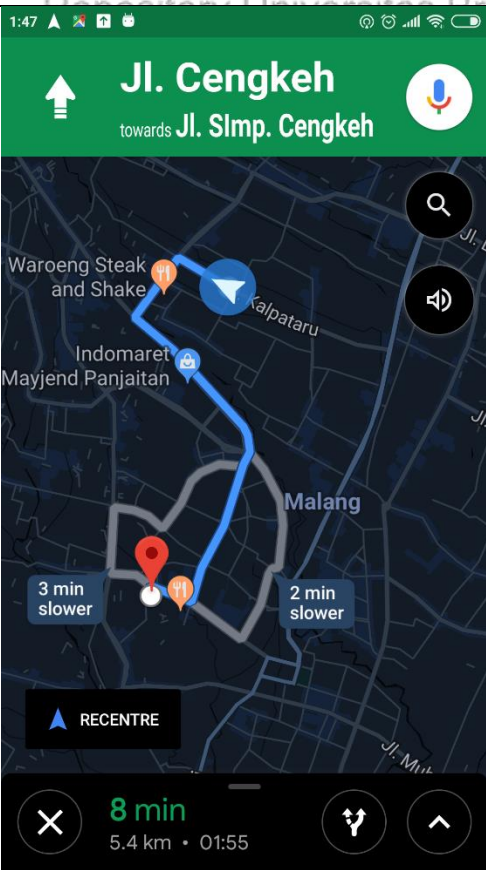
<p>Tujuan Pengujian</p>	<p>Untuk memberikan kepastian pada aplikasi yang dikembangkan dalam melihat deskripsi dari tiap tempat oleh-oleh</p>
<p>Prosedur Pengujian</p>	<ol style="list-style-type: none"> 1. Membuka aplikasi rekomendasi oleh-oleh. 2. Berada pada halaman utama dari rekomendasi oleh-oleh. 3. Bobot yang tersedia dimasukkan sesuai kriteria pengguna. 4. Menekan tombol dibawah berupa next untuk melanjutkan kehalaman berikutnya 5. Mendapatkan beberapa rekomendasi oleh-oleh yang disediakan oleh system. 6. Memilih rekomendasi oleh-oleh yang ditampilkan
<p>Hasil yang Diharapkan</p>	<p>Pengguna melihat deskripsi dari hasil rekomendasi</p>
<p>Screenshot</p>	 <p>Gambar 6.4 Hasil Pengujian Melihat detail hasil rekomendasi</p>

Tabel 6.3 adalah penjelasan bagaimana fungsionalitas dari melihat detail hasil rekomendasi diuji.

Tabel 6.4 Pengujian Fungsional Melihat Lokasi Hasil Rekomendasi

<p>Nomor Kasus Uji</p>	<p>TOB-04</p>
<p>Nama Kasus Uji</p>	<p>Melihat Lokasi Hasil Rekomendasi</p>
<p>Objek Uji</p>	<p>Kebutuhan Fungsional</p>

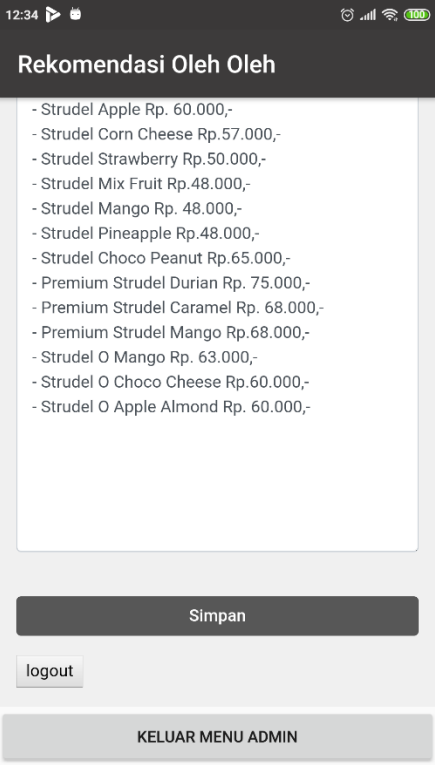


Tujuan Pengujian	Untuk memberikan kepastian pada aplikasi yang dikembangkan dalam melihat lokasi tempat oleh-oleh yang tersedia
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka aplikasi rekomendasi oleh-oleh. 2. Berada pada halaman utama dari rekomendasi oleh-oleh. 3. Bobot yang tersedia dimasukkan sesuai kriteria pengguna. 4. Menekan tombol dibawah berupa next untuk melanjutkan kehalaman berikutnya 5. Mendapatkan beberapa rekomendasi oleh-oleh yang disediakan oleh system. 6. Memilih rekomendasi oleh-oleh yang ditampilkan 7. Menekan gambar yang berupa seperti peta pada rekomendasi oleh-oleh yang terpilih
Hasil yang Diharapkan	Pengguna dapat melihat lokasi dari rekomendasi tempat oleh-oleh
Screenshot	 <p>Gambar 6.5 Hasil Pengujian Melihat lokasi tempat rekomendasi</p>

Tabel 6.4 adalah penjelasan bagaimana fungsionalitas dari melihat lokasi dari tempat rekomendasi.



Tabel 6.5 Pengujian Fungsional Melihat Lokasi Hasil Rekomendasi

Nomor Kasus Uji	TOB-05
Nama Kasus Uji	Mengubah Detail Tempat Oleh-oleh
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memberikan kepastian pada aplikasi yang dikembangkan dalam mengubah detail dari tempat oleh-oleh
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka aplikasi rekomendasi oleh-oleh. 2. Berada pada halaman utama dari rekomendasi oleh-oleh. 3. Memilih tombol admin. 4. Melakukan Login. 5. Mengubah data dari tempat oleh-oleh sesuai dengan tujuan masing masing. 6. Memilih tombol simpan jika sudah sesuai dengan kebutuhan.
Hasil yang Diharapkan	Pengguna yaitu admin dapat mengubah data detail tempat oleh-oleh
Screenshot	 <p>The screenshot shows a mobile application interface titled 'Rekomendasi Oleh Oleh'. It displays a list of products with their prices: Strudel Apple (Rp. 60.000,-), Strudel Corn Cheese (Rp. 57.000,-), Strudel Strawberry (Rp. 50.000,-), Strudel Mix Fruit (Rp. 48.000,-), Strudel Mango (Rp. 48.000,-), Strudel Pineapple (Rp. 48.000,-), Strudel Choco Peanut (Rp. 65.000,-), Premium Strudel Durian (Rp. 75.000,-), Premium Strudel Caramel (Rp. 68.000,-), Premium Strudel Mango (Rp. 68.000,-), Strudel O Mango (Rp. 63.000,-), Strudel O Choco Cheese (Rp. 60.000,-), and Strudel O Apple Almond (Rp. 60.000,-). At the bottom of the list, there is a 'Simpan' button, a 'logout' button, and a 'KELUAR MENU ADMIN' button.</p>

Gambar 6.6 Hasil Pengujian Mengubah Detail Tempat oleh-oleh

Tabel 6.5 adalah penjelasan bagaimana fungsionalitas dari mengubah detail tempat dari oleh-oleh.



Dari hasil yang telah diuji pada black box tersebut, telah didapatkan bahwa semua fungsionalitas sudah valid dan memenuhi kebutuhan yang sudah rancang sebelumnya.

Untuk menentukan nilai fungsionalitas dari pengujian *black box*, penghitungan dari nilai tersebut dapat dilihat dibawah ini:

$$\begin{aligned} \text{fungsionalitas} &= \frac{\sum \text{fungsional valid}}{\sum \text{daftar kebutuhan}} \\ &= \frac{5}{5} \times 100\% \\ &= 100\%. \end{aligned}$$

6.2 Pengujian System Usability Scale

Pengujian non-fungsional *System Usability Scale* (SUS) dilakukan untuk menguji sebuah system atau aplikasi dengan jumlah responden yaitu 5 orang. Karakteristik dari 5 responden tersebut adalah mahasiswa dengan umur 18 – 20 tahun dengan pengalaman menggunakan perangkat bergerak lebih dari 5 tahun.

Penilaian SUS dari responden yang tertera ditunjukkan pada Tabel 4.8.

Tabel 6.6 Pengujian SUS

Responden	Hasil Penilaian	Skor
1	33 x 2.5	82.5
2	31 x 2.5	77.5
3	34 x 2.5	85
4	26 x 2.5	65
5	27 x 2.5	67.5
Rerata		$377.5 / 5 = 75.5$

Hasil dari SUS dengan 5 responden dengan 10 pertanyaan dicari reratanya dan dibagi dengan jumlah respondennya yaitu 5, mendapatkan hasil seperti berikut:

$$\begin{aligned} \text{SUS} &= \frac{377.5}{5} \\ &= \frac{377.5}{5} \\ &= 75.5 \text{ (acceptable)} \end{aligned}$$



Mengacu pada Gambar 2.1, hasil dari SUS masuk kedalam *range acceptable* dimana nilai dari sus lebih dari 60 dan mendapatkan grade B karena nilai diatas 73 sampai 80,3.

6.3 Pengujian Validasi Algoritme

Pengujian non-fungsionalitas yang kedua yaitu validasi algoritme, dimana algoritme akan dilakukan pengujian dengan perhitungan manual dan perhitungan sistem untuk mengecek alogritme tersebut membuahakan hasil yang sama atau tidak. Pengujian tersebut menyesuaikan pada bab perancangan di Tabel 4.8.

Diketahui titik dari pengguna yaitu -7.947421 sebagai garis lintang dan 112.6243284 sebagai garis bujur dan data alternatif yang dipakai yaitu merupakan data-data yang tersedia pada basis data yang telah ada dengan isi bobot dari pengisian pengguna sesuai pada Tabel 6.6 berikut. Bobot pada tabel 6.6 merupakan bobotan input yang diimplikasikan sebagai sampel dengan memasukkan nilai kriteritia sebagai prioritas penilaian.

Tabel 6.6 Bobot Kriteria Pengujian

Kolom	Bobot
Harga (Rp)	0.5
Kadaluarsa (Hari)	0.2
Jarak (Km)	0.3

Data tempat oleh-oleh tersebut atau disebut dengan alternatif memiliki nilai-nilai kriteria yang berbeda beda dari harga dengan satuan rupiah, kadaluarsa dengan satuan hari dan jarak menggunakan satuan kilometer. Jarak tersebut telah dihitung dari tempat pengguna berada sampai dengan tempat oleh-oleh yang dipaparkan. Data tersebut sesuai dengan Tabel 6.7.

Tabel 6.7 Alternatif Data Pengujian

No	nama	Harga (Rp)	Kadaluarsa (Hari)	Jarak (Km)
1	Malang Strudel Sukarno Hatta 2	48000	5	0.753730003
2	Malang Strudel Sukarno Hatta 1	48000	5	1.087603359
3	Lapis Malang Kawi	30000	5	3.208660833
4	Pia Cap Mangkok	11500	12	3.249444917
5	Pia Cap Mangkok Semeru	17000	10	2.116433731



6	Goedang Oleh-oleh	11500	12	1.060574858
7	Malang Strudel Singosari	48000	5	9.819900028
8	Kripik Tempe Rizky Barokah	5000	11	2.618772863
9	Apple Strudle	50000	7	1.431100508
10	Mimi Apple Pie	20000	11	13.47832695
11	Queen Apple	60000	12	3.194073858
12	Amanda Brownies	35999	13	3.968452858
13	Lancar Jaya Tempe and Fruit Chips	15000	20	2.274815391
14	Lapis Kukus Tugu Malang Siguragura	25000	30	2.317606738
15	Lapis Kukus Tugu Malang	25000	30	2.279309441

Tabel 6.8 dibawah ini membandingkan nilai yang dikeluarkan dari sistem dan perhitungan manual.

Tabel 6.8 Perbandingan Hasil Antara Dua Perhitungan

No	Keluaran Sistem		Perhitungan Manual	
	Alternatif	Hasil	Alternatif	Hasil
1	Malang Strudel Sukarno Hatta 2	0.721451062	Malang Strudel Sukarno Hatta 2	0.721451
2	Malang Strudel Sukarno Hatta 1	0.715646558	Malang Strudel Sukarno Hatta 1	0.715647
3	Lapis Malang Kawi	0.69577587	Lapis Malang Kawi	0.695776
4	Pia Cap Mangkok	0.76051635	Pia Cap Mangkok	0.760516
5	Pia Cap Mangkok Semeru	0.792153778	Pia Cap Mangkok Semeru	0.792154
6	Goedang Oleh-oleh	0.842444506	Goedang Oleh-oleh	0.842445



7	Malang Strudel Singosari	0.269078138	Malang Strudel Singosari	0.269078
8	Kripik Tempe Rizky Barokah	0.791613173	Kripik Tempe Rizky Barokah	0.791613
9	Apple Strudle	0.70991304	Apple Strudle	0.709913
10	Mimi Apple Pie	0.211817326	Mimi Apple Pie	0.211817
11	Queen Apple	0.640519408	Queen Apple	0.640519
12	Amanda Brownies	0.674291408	Amanda Brownies	0.674291
13	Lancar Jaya Tempe and Fruit Chips	0.846011377	Lancar Jaya Tempe and Fruit Chips	0.846011
14	Lapis Kukus Tugu Malang Siguragura	0.83847448	Lapis Kukus Tugu Malang Siguragura	0.838474
15	Lapis Kukus Tugu Malang	0.84037876	Lapis Kukus Tugu Malang	0.840379

Dari hasil pada Tabel 6.8, diketahui bahwa nilai dari perhitungan manual dengan sistem memiliki perbedaan yang tidak jauh. Dengan begitu hasil pengujian validasi algoritme tersebut adalah:

$$\begin{aligned}
 \text{valid} &= \frac{\text{Jumlah data sama}}{\text{jumlah semua data}} \times 100\% \\
 &= \frac{15}{15} \times 100\% \\
 &= 100\%
 \end{aligned}$$



BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Selesaiannya penelitian yang telah dikerahkan, didapatkan beberapa kesimpulan yang dapat memenuhi keseluruhan penelitian. Dimana kesimpulan tersebut yaitu:

1. Dengan ada aplikasi untuk mencari rekomendasi oleh-oleh, pengguna dapat mudah mencari tempat oleh-oleh dimana terdapat kriteria jarak, harga dan kadaluarsa untuk menunjang pencarian sesuai dengan keinginan pengguna. Disediakan juga tampilan yaitu menu dan juga apakah tempat tersebut buka atau tutup sesuai dengan inputin dari admin masing masing tempat oleh-oleh.
2. Sistem rekomendasi ini menggunakan metode TOPSIS sebagai algoritme rekomendasi dengan metode perangkat lunak yaitu waterfall. Kriteria yang digunakan pada metode TOPSIS dan aplikasi yang dibuat ini yaitu jarak, harga dan kadaluarsa yang dirancangan menggunakan diagram seperti use case sebagai guide line dari pembuatan aplikasi.
3. Metode TOPSIS merupakan salah satu metode untuk melakukan pencarian rekomendasi dimana TOPSIS ini memiliki tingkat kompleksitas yang rendah dan juga memiliki *error rate* rendah
4. Sistem rekomendasi ini menggunakan android studio untuk melakukan implementasi dari pengembangannya. Untuk algoritme yang dipakai guna pencarian tempat oleh-oleh, menggunakan metode TOPSIS dimana tiap tempat dilakukan perangkangan untuk mencari hasil yang sesuai dengan kriteria pengguna.
5. Sistem rekomendasi tempat oleh oleh khas malang memberikan hasil pengujian blackbox bernilai valid (100% *completion rate*). Untuk pengujian algoritme, nilai dari perhitungan manual dengan sistem memiliki perbedaan sedikit dan tidak mempengaruhi perangkangan pada hasil keluar rekomendasi tempat oleh-oleh. Pengujiannya dengan *System Usability Scale* menghasilkan nilai yaitu 75,5 dimana nilai tersebut termasuk dalam range *acceptable* atau dapat diterima oleh pengguna.

7.2 Saran

Beberapa saran dikerahkan oleh penulis saat penelitian ini selesai dikembangkan. Dengannya saran yang disampaikan ini, pengembangan rekomendasi oleh-oleh ini dapat dilanjutkan sebagaimana kekurangan tersebut dapat dipenuhi secara riil. Saran dari peneliti yaitu sebagai berikut:

1. Alangkah baiknya mengidentifikasi kriteria selain dari tiga kriteria yang telah dikerahkan untuk memberikan rekomendasi yang lebih ideal.
2. Dikarenakan penggunaan dari sensor GPS yang masih dibawah dari optimal, disarankan untuk melakukan optimasi dari penggunaan GPS untuk menghat sumber daya yang dipakai pada perangkat bergerak.

DAFTAR REFERENSI

- Alshamrani, A. dan Bahattab, A., 2015. *A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model*. International Journal of Computer Science Issues (IJCSI), 12(1), p.106.
- Baumgartner, Juergen, et al. 2019. *Pictorial System Usability Scale (P-SUS): Developing an Instrument for Measuring Perceived Usability*. University of Fribourg. Tersedia di: <<https://dl.acm.org/doi/pdf/10.1145/3290605.3300299?>> [Diakses 11 Juni 2020]
- Crockford, D. 2015. *Json*. Tersedia di: <https://github.com/douglascrockford/JSON-js/blob/master/README>. [Diakses 20 September 2018].
- Developer, Android, 2018. *Platform Architecture*. [Online] Developers Android. Tersedia di: <<https://developer.android.com/guide/platform/>>[Diakses 21 Desember 2019]
- Dewi, R.K., Hanggara, B.T.; Pinandito, A., 2018. *A Comparison Between AHP and Hybrid AHP for Mobile Based Culinary Recommendation System*. International Journal of Interactive Mobile Technologies. Tersedia di: <<http://journals.sfu.ca/onlinejour/index.php/ijim/article/view/7561>> [Diakses 13 September 2018].
- Ependi, Usman, Panjaitan, Febriyanti, Hutrianto, 2017. *System Usability Scale Antarmuka Palembang Guide Sebagai Media Pendukung Asian Games XVIII*. Universitas Binda Darma.
- Firman, Asfar, A.M. Irfan Taufan, dan Suhardiman. 2019. *Implementasi Media Pembelajaran Berbasis Android Sebagai Sistem Fast Respond Evaluation*. STKIP Muhammadiyah Bone. Tersedia di: <<http://jurnal.poliupg.ac.id/index.php/snp2m/article/viewFile/1935/1778>> [Diakses 11 Juni 2019].
- Garmin, GPS, 2019. Apa Itu GPS?. [Online] Garmin. Tersedia di: <<https://www.garmin.co.id/about-gps/>>[Diakses 21 Desember 2019]
- Ghanimata, F. dan Kamal, M. 2012. Analisis pengaruh harga, kualitas produk, dan lokasi terhadap keputusan pembelian (Studi pada Pembeli Produk Bandeng Juwana-Elrina Semarang). Universitas Diponegoro.
- 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. STMIK Mikroskil.
- Hibatullah, Muhamad Hilmi, Dewi, R.K., Marji, Marji. 2018. Implementasi TOPSIS Pada Sistem Rekomendasi Tempat Wisata Pantai Di Sekitar Malang Berbasis Lokasi. Universitas Brawijaya.



KBBI, 2016. Oleh – Oleh. [Online] Tersedia di: <<https://kbbi.kemdikbud.go.id/entri/oleh-oleh>> [Diakses 13 September 2018].

Komalasari, Nia. 2016. *Sistem Pendukung Keputusan Kelaikan Terbang (SPK2T)*. Universitas Nurtanio Bandung. Tersedia di: <<http://jurnal.unnur.ac.id/index.php/indept/article/view/134/105>>

Kramer, M., 2018. *Best Practices in Systems Development Lifecycle: An Analyses Based on the Waterfall Model*. Review of Business & Finance Studies, 9(1), pp.77-84.

Magdalena, Hilyah. 2012. Sistem Pendukung Keputusan Untuk Menentukan Mahasiswa Lulusan Terbaik di Perguruan Tinggi (Studi Kasus STMIK Atma Luhur Pangkalpinang). STMIK Atma Luhur Pangkalpinang.

MeasuringU, 2020. *System Usability Scale*. [Online] Tersedia di: <<https://measuringu.com/sus/>> [Diakses 12 Juni 2020].

Purnawan, S.I., Marisa, F. dan Wijaya, I.D. 2018. Aplikasi Pencarian Pariwisata Dan Tempat Oleh-Oleh Terdekat Menggunakan Metode Haversine Berbasis Android. JIMP-Jurnal Informatika Merdeka Pasuruan, 3(2). Nurhayati, Asti. 2010. *Analisis Pengujian Perangkat Lunak Augmented Reality*. Indonesia.

Purba, Krisna Satria Pranata. 2016. Analisis Faktor-Faktor Yang Dipertimbangkan Oleh Konsumen Dalam Pengambilan Keputusan Pembelian Keripik Tempe, di Kota Malang. Universitas Brawijaya.

Purwandani, A.R., Husodo, A.Y. dan Bimantoro, F. 2019. *Analisis Efektifitas Metode Weighted Product dan TOPSIS dalam Mendiagnosa Serangan Asma*. Journal of Computer Science and Informatics Engineering (J-Cosine) - Tersedia di: <<http://jcosine.if.unram.ac.id/index.php/jcosine/article/view/185/33>>. [Diakses 2 Desember 2019].

Rohayani, Hetty. 2013. Analisis Sistem Pendukung Keputusan dalam Memilih Program Studi Menggunakan Metode Logika Fuzzy. STIKOM Dinamika Bangsa Jambi.

Rosi, Luki Hamzah. 2016. Implementasi Metode Fuzzy Tsukamoto Untuk Pendukung Keputusan Jumlah Produksi Barang Pada Distro Anime. Universitas Nusantara PGRI Kediri.

Sari, Wheny Mela, et al. 2019. *Evaluasi pada Website Fakultas Peternakan Universitas Brawijaya menggunakan Usability Testing*. Universitas Brawijaya. Tersedia di: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5706/2725>> [Diakses 12 Juni 2020].

Sawel, F. D., Sinsuw, A. A. E., Putro, M. D., 2016. *Sistem Pendukung Keputusan Rekomendasi Makanan Khas Sulawesi Utara yang Menunjang Diet*. Universitas Sam Ratulangi. Tersedia di: <<https://ejournal.unsrat.ac.id/index.php/informatika/article/view/13485>>. [Diakses 2 Januari 2019].



Subakti, Irfan. 2002. Sistem Pendukung Keputusan (Decision Support System). Surabaya: Jurusan Teknik Informatika FTI, Institut Teknologi Sepuluh Nopember.

Wibisono, Waskitho, Baskoro, Fajar. 2002. Pengujian Perangkat Lunak dengan Menggunakan Model Behaviour UML. Institut Teknologi Sepuluh November.

Wulandari, Denok dan Willyansah, Willyansah. 2020. *Sistem Penunjang Keputusan (Spk) Dalam Penjualan Komputer Pada Cv. Radian Grup Dengan Menerapkan Bahasa Pemrograman Java*. LPPM AMIK Mitra Gama. Tersedia di <<http://ojsamik.amikmitragama.ac.id/index.php/js/article/view/67>> [Diakses 2 Januari 2020].



LAMPIRAN A PENGUJIAN SYSTEM USABILITY SCALE

A.1 Responden Satu

Nama Pengguna: Faishal P.A.

Umur Pengguna: 22

System Usability Scale

Instruksi: Pada tiap pernyataan, beri tanda ceklis (✓) sesuai dengan reaksi pada aplikasi Sistem Rekomendasi Oleh Oleh.

NO	Pernyataan	Tidak Setuju			Sangat Setuju
1	Saya akan sering menggunakan aplikasi ini			✓	
2	Saya melihat aplikasi ini tidak terlalu kompleks				✓
3	Saya merasa aplikasi ini mudah untuk digunakan			✓	
4	Saya merasa membutuhkan bantuan teknis untuk menggunakan aplikasi ini	✓			
5	Saya merasa fungsi fungsi yang ada dalam aplikasi ini terintegrasi dengan baik			✓	
6	Saya merasa aplikasi ini tidak konsisten		✓		
7	Saya membayangkan bahwa banyak orang akan belajar secara cepat untuk menggunakan aplikasi ini			✓	
8	Saya melihat aplikasi ini tidak praktis untuk digunakan		✓		
9	Saya merasa percaya diri dalam menggunakan aplikasi ini				✓
10	Saya harus belajar lebih banyak lagi sebelum menggunakan aplikasi ini		✓		

A.2 Responden Dua

Nama Pengguna: Arvega Rachmat

Umur Pengguna: 22

System Usability Scale

Instruksi: Pada tiap pernyataan, beri tanda ceklis (✓) sesuai dengan reaksi pada aplikasi Sistem Rekomendasi Oleh Oleh.

NO	Pernyataan	Tidak Setuju			Sangat Setuju
1	Saya akan sering menggunakan aplikasi ini				✓
2	Saya melihat aplikasi ini tidak terlalu kompleks			✓	
3	Saya merasa aplikasi ini mudah untuk digunakan				✓
4	Saya merasa membutuhkan bantuan teknis untuk menggunakan aplikasi ini	✓		✓	
5	Saya merasa fungsi fungsi yang ada dalam aplikasi ini terintegrasi dengan baik			✓	
6	Saya merasa aplikasi ini tidak konsisten		✓		
7	Saya membayangkan bahwa banyak orang akan belajar secara cepat untuk menggunakan aplikasi ini			✓	
8	Saya melihat aplikasi ini tidak praktis untuk digunakan		✓		
9	Saya merasa percaya diri dalam menggunakan aplikasi ini				✓
10	Saya harus belajar lebih banyak lagi sebelum menggunakan aplikasi ini	✓			



A.3 Responden Tiga

Nama Pengguna: Toruna Pajek

Umur Pengguna: 29

System Usability Scale

Instruksi: Pada tiap pernyataan, beri tanda ceklis (✓) sesuai dengan reaksi pada aplikasi Sistem Rekomendasi Oleh Oleh.

NO	Pernyataan	Tidak Setuju			Sangat Setuju.
1	Saya akan sering menggunakan aplikasi ini			✓	
2	Saya melihat aplikasi ini tidak terlalu kompleks				✓
3	Saya merasa aplikasi ini mudah untuk digunakan			✓	
4	Saya merasa membutuhkan bantuan teknisi untuk menggunakan aplikasi ini				✓
5	Saya merasa fungsi fungsi yang ada dalam aplikasi ini terintegrasi dengan baik				✓
6	Saya merasa aplikasi ini tidak konsisten				✓
7	Saya membayangkan bahwa banyak orang akan belajar secara cepat untuk menggunakan aplikasi ini			✓	
8	Saya melihat aplikasi ini tidak praktis untuk digunakan				✓
9	Saya merasa percaya diri dalam menggunakan aplikasi ini			✓	
10	Saya harus belajar lebih banyak lagi sebelum menggunakan aplikasi ini				✓

A.4 Responden Empat

Nama Pengguna: Ahmad Fareza Y

Umur Pengguna: 21

System Usability Scale

Instruksi: Pada tiap pernyataan, beri tanda ceklis (✓) sesuai dengan reaksi pada aplikasi Sistem Rekomendasi Oleh Oleh.

NO	Pernyataan	Tidak Setuju			Sangat Setuju
1	Saya akan sering menggunakan aplikasi ini				✓
2	Saya melihat aplikasi ini tidak terlalu kompleks				✓
3	Saya merasa aplikasi ini mudah untuk digunakan			✓	
4	Saya merasa membutuhkan bantuan teknisi untuk menggunakan aplikasi ini			✓	
5	Saya merasa fungsi fungsi yang ada dalam aplikasi ini terintegrasi dengan baik		✓		
6	Saya merasa aplikasi ini tidak konsisten			✓	
7	Saya membayangkan bahwa banyak orang akan belajar secara cepat untuk menggunakan aplikasi ini				✓
8	Saya melihat aplikasi ini tidak praktis untuk digunakan		✓		
9	Saya merasa percaya diri dalam menggunakan aplikasi ini				✓
10	Saya harus belajar lebih banyak lagi sebelum menggunakan aplikasi ini		✓		



A.5 Responden Lima

Nama Pengguna: *Erwin Haryo Daris*

Umur Pengguna: 21

System Usability Scale

Instruksi: Pada tiap pernyataan, beri tanda ceklis (✓) sesuai dengan reaksi pada aplikasi Sistem Rekomendasi Oleh Oleh.

NO	Pernyataan	Tidak Setuju			Sangat Setuju
1	Saya akan sering menggunakan aplikasi ini		✓		
2	Saya melihat aplikasi ini tidak seharusnya sekompleks ini				✓
3	Saya merasa aplikasi ini mudah untuk digunakan			✓	
4	Saya merasa membutuhkan bantuan teknis untuk menggunakan aplikasi ini			✓	
5	Saya merasa fungsi fungsi yang ada dalam aplikasi ini terintegrasi dengan baik			✓	✓
6	Saya merasa aplikasi ini tidak konsisten				✓
7	Saya membayangkan bahwa banyak orang akan belajar secara cepat untuk menggunakan aplikasi ini	✓			
8	Saya melihat aplikasi ini tidak praktis untuk digunakan			✓	
9	Saya merasa percaya diri dalam menggunakan aplikasi ini		✓		
10	Saya harus belajar lebih banyak lagi sebelum menggunakan aplikasi ini				✓

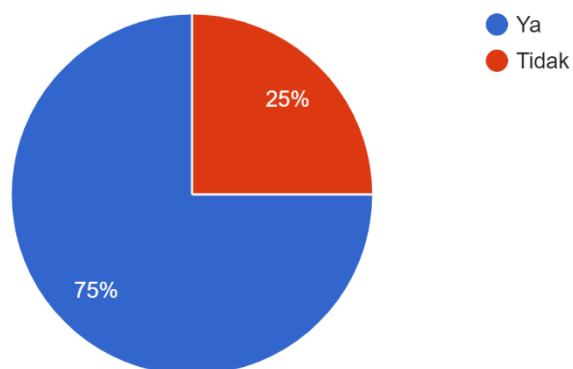


LAMPIRAN B KUESIONER KEBUTUHAN

B.1 Gambar Satu

Apakah kamu gemar membelikan oleh oleh untuk keluarga maupun kerabat?

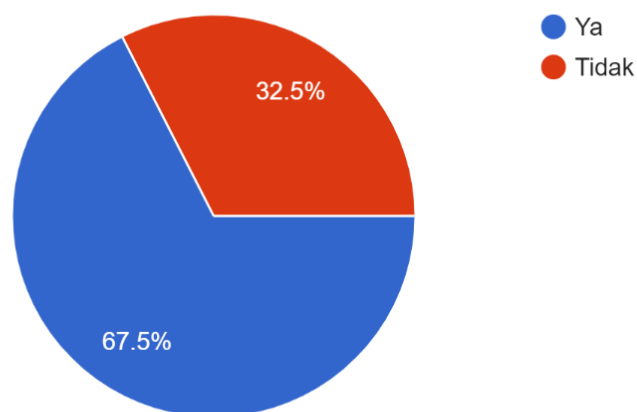
40 responses



B.2 Gambar Dua

Apakah kamu pernah merasa bimbang dalam mencari sebuah oleh oleh?

40 responses

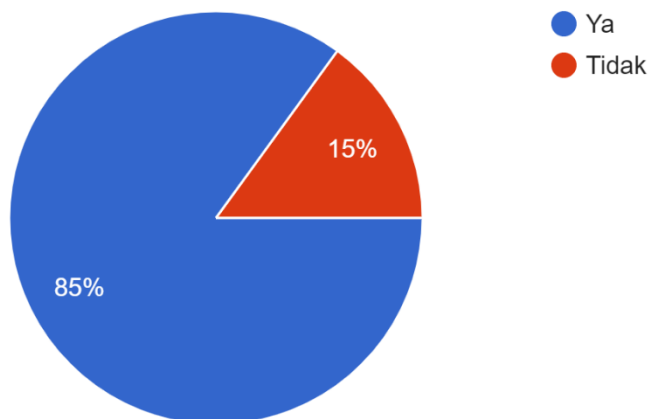




B.3 Gambar Tiga

Apakah harga menjadi salah satu pertimbangan anda dalam mencari oleh oleh?

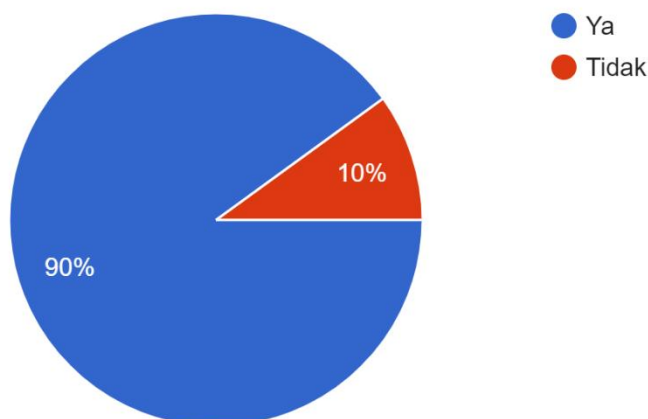
40 responses



B.4 Gambar Empat

Apakah kadaluarsa dari sebuah produk oleh oleh menjadi salah satu pertimbangan anda m...lih oleh oleh yang terjangkau?

40 responses

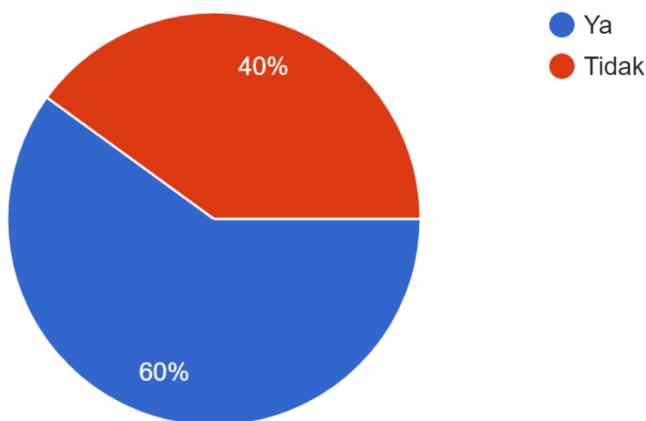




B.5 Gambar Lima

Apakah jarak anda dengan tujuan anda mendapatkan oleh oleh menjadi hal yang diperhatikan?

40 responses





LAMPIRAN C HASIL KELUARAN ALGORITME

C.1 JSON Hasil Penghitungan Algoritme

```

[
  {
    "id": "13",
    "name": "Lancar Jaya Tempe and Fruit Chips",
    "latitude": "-7.9602189",
    "longitude": "112.640444",
    "harga": "15000",
    "kadaluarsa": "20",
    "menu": "- Apple Strudle Rp 50000,- - Chocolate Strudle Rp. 56.000,-",
    "url": "http://topsisfhj.xyz/TOPSIS_OLEH_OLEH/gambar/13.jpg"
  },
  {
    "id": "6",
    "name": "Pia Cap Mangkok Soehat",
    "latitude": "-7.9388178",
    "longitude": "112.6284863",
    "harga": "11500",
    "kadaluarsa": "12",
    "menu": "- 1 pack isi 5 biji Rp11.500 - 1 dus isi 12 biji Rp26.500 - 1 dus isi 25 biji (kacang hijau) Rp49.000 - 1 dus isi 25 biji (cokelat) Rp49.000 - 1 dus isi 25 biji (kopi) Rp49.000 - 1 dus isi 25 biji (tangwe) Rp53.000 - 1 dus isi 25 biji (keju) Rp53.000",
    "url": "http://topsisfhj.xyz/TOPSIS_OLEH_OLEH/gambar/6.jpg"
  },
  {
    "id": "15",
    "name": "Lapis Kukus Tugu Malang Siguragura",
    "latitude": "-7.9643987",
    "longitude": "112.6127303",
    "harga": "25000",
    "kadaluarsa": "30",
    "menu": "- Lapis Kukus Original Rp. 25.000- Lapis Kukus Chocopandan Rp. 25.000- Lapis Kukus Brownies Coklat Rp. 28.000- Lapis Kukus Brownies Keju Rp. 28.000",
    "url": "http://topsisfhj.xyz/TOPSIS_OLEH_OLEH/gambar/15.jpg"
  }
]
    
```