

**RANCANG BANGUN APLIKASI REKOMENDASI LOKASI WIFI
HOTSPOT BERBASIS ANDROID MENGGUNAKAN METODE
WEIGHTED PRODUCT**

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

RIZAL BAGUS RAHMADHANI

NIM. 105060800111050

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

**RANCANG BANGUN APLIKASI REKOMENDASI LOKASI WIFI
HOTSPOT BERBASIS ANDROID MENGGUNAKAN METODE
WEIGHTED PRODUCT**

SKRIPSI

Untuk Memenuhi Sebagian Persyaratan Memperoleh Gelar Sarjana Komputer



Disusun Oleh:

RIZAL BAGUS RAHMADHANI

NIM. 105060800111050

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Aryo Pinandito, ST., M.MT.

NIK. 83051916110374

Aswin Suharsono, ST., MT.

NIK. 84091906110251

LEMBAR PENGESAHAN
RANCANG BANGUN APLIKASI REKOMENDASI LOKASI WIFI
HOTSPOT BERBASIS ANDROID MENGGUNAKAN METODE
WEIGHTED PRODUCT

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

RIZAL BAGUS RAHMADHANI

NIM. 105060800111050

Skripsi ini telah diuji dan dinyatakan lulus tanggal 9 Januari 2015

Penguji I

Penguji II

Dr. Eng Herman Tolle, ST., MT.
NIP. 19740823 200012 1 001

Issa Arwani, S.Kom, M.Sc.
NIP. 19830922 201212 1 003

Penguji III

Fajar Pradana, S.ST, M.Eng.
NIK. 871121 16110371

Mengetahui

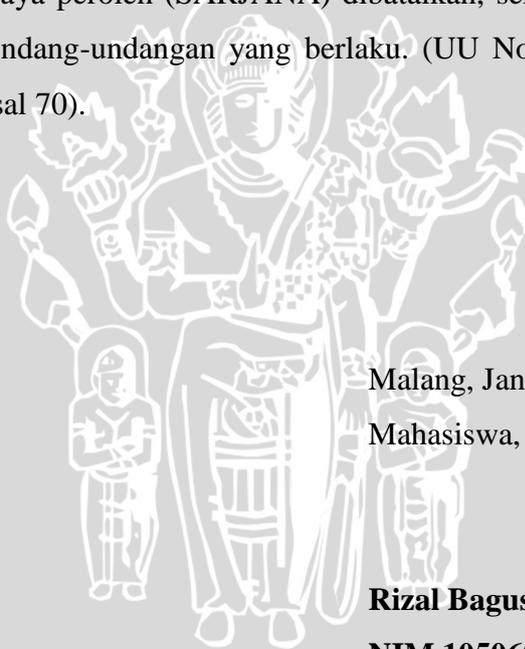
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, MT.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

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, Januari 2015

Mahasiswa,

Rizal Bagus Rahmadhani

NIM 105060800111050

KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Allah SWT karena atas rahmat dan karunia-Nya, saya dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Rekomendasi Lokasi WiFi *Hotspot* Berbasis Android Menggunakan Metode *weighted product*” sebagai salah satu persyaratan untuk menyelesaikan studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Tak lupa shalawat serta salam penulis haturkan kepada junjungan besar, Nabi Muhammad SAW.

Penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi, diantaranya:

1. Bapak Moh. Thamrin, Ibu S. Ch. Hamidah, Reza Bagus A., Rindha Ayu A., dan Muhammad Firdaus Firmansyah serta seluruh keluarga atas segala nasehat, kasih sayang, perhatian dan kesabarannya dalam mendidik penulis serta memberikan doa dan semangat secara terus-menerus demi terselesaikannya skripsi ini.
2. Bapak Aryo Pinandito, S.T., M.MT. selaku dosen pembimbing I dan Bapak Aswin Suharsono, S.T. M.T. selaku dosen pembimbing II yang telah memberikan ilmu serta saran yang luar biasa untuk menyelesaikan skripsi ini.
3. Bapak Budi Darma, S.T., M.T. selaku dosen pembimbing akademik yang dengan sabar memberikan pengarahan selama penulis menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Seluruh dosen dan karyawan Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Novrizal Putra Aulia, S.T., Agung Rudiardi, S.Ked., Agung Rudiarto, S.H., Ahmad Baihakki, S.ST., Dika Rizky Yuniyanto, S.Kom., dan Nailah Husna,

S.Kom. atas dukunganya secara teknis maupun moral.

6. Teman-teman seperjuangan (Awangga, Geby, Abe, Helos, Nena, Citra, Tacik, Rey, dan Irwan)
7. Sahabat-sahabat Informatika, PTIIK, LPM DISPLAY PTIIK, serta semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah memberikan doa, bantuan serta semangat kepada penulis selama menempuh studi di Teknik Informatika Universitas Brawijaya.

Semoga Tugas Akhir ini dapat memberikan manfaat bagi pembaca sekaligus dapat menjadi bahan acuan untuk penelitian selanjutnya.

Malang, Januari 2015

Penulis



ABSTRAK

Internet merupakan suatu kebutuhan dalam berkegiatan dan transaksi. Penggunaan Internet pada banyak hal seperti pengiriman data dalam pekerjaan hingga sekedar streaming video dan bermain permainan online untuk hiburan menjadikan Internet menjadi sebuah kebutuhan. Fasilitas WiFi di tempat umum dapat menjadi alternative lokasi bagi pengguna internet. Namun tidak sembarang tempat menjadi pilihan para pengguna Internet. Maka dibutuhkan sebuah aplikasi rekomendasi lokasi WiFi yang dapat membantu pengguna dalam memilih lokasi yang mereka inginkan. Pada penelitian ini, aplikasi rekomendasi lokasi WiFi berbasis Android dikembangkan dengan metode *weighted product* dan rumus Haversine. Rumus Haversine menghitung jarak berdasarkan *longitude* dan *latitude*. Sedangkan metode *weighted product* digunakan untuk rekomendasi lokasi WiFi berdasarkan jarak lokasi, nilai kenyamanan lokasi, harga minimal, dan kecepatan Internet lokasi. Selain itu, aplikasi ini juga memberikan informasi tentang lokasi-lokasi tersebut.

Berdasarkan pengujian yang dilakukan, keseluruhan fungsional aplikasi ini dapat berjalan dengan baik dan metode yang *weighted product* dapat diterapkan pada aplikasi ini. Selain itu aplikasi ini memiliki nilai *usability* sebesar 88.6% yang menunjukkan bahwa aplikasi ini dapat membantu pengguna dalam memilih lokasi WiFi yang diinginkan.

Kata Kunci: Aplikasi Rekomendasi, *weighted product*, Haversine, WiFi

ABSTRACT

Internet is a necessity in doing activities and transactions. The use of Internet on many things such as data transmission in the job or stream video and play online game for entertainment make Internet becomes a necessity. WiFi facility in public room can be an alternative location for Internet users. But not just any places be the Internet users choice. So WiFi location app recommendations is needed to assist Internet users in selecting their desired location. In this study, WiFi location-based applications on Android was developed by the method of weighted product and Haversine formula. Haversine formula gives the distance from the calculation based in the longitude and latitude. While the weighted product method will provide recommendations based on locations distance, locations convenience value, minimum price, and speed of Internet. In addition, this application also provides information on these sites.

Based on testing, the overall functional applications can be run well and the weighted product method can be applied to this application. In addition, this application has a value of 88.65 on a usability test that shows this application can help the user to choose the desired WiFi locations.

KEYWORDS: APPLICATION RECCOMENDATIONS, WEIGHTED PRODUCT, HAVERSINE, WIFI

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Rumus Haversine.....	5
2.2 Metode <i>Weighted Product</i>	5
2.3 AsyncTask	7
2.4 <i>Usability Testing</i>	7
2.4.1 Kuesioner USE	8



2.5 Skala Likert..... 9

BAB III METODOLOGI PENELITIAN 10

3.1 Metodologi Penelitian..... 10

3.1.1 Pengumpulan Data..... 10

3.1.2 Studi Literatur..... 14

3.1.3 Analisis Kebutuhan..... 14

3.1.4 Perancangan..... 15

3.1.5 Implementasi 16

3.1.6 Pengujian dan Analisis 16

BAB IV PERANCANGAN..... 17

4.1 Deskripsi Umum Aplikasi 17

4.2 Analisis Kebutuhan..... 18

4.2.1 Analisis Aktor..... 18

4.2.2 Kebutuhan Fungsional..... 18

4.2.3 Diagram *Use case*..... 19

4.2.4 Kebutuhan Non Fungsional..... 23

4.3 Perancangan Perangkat Lunak..... 23

4.3.1 Mendapatkan Lokasi Pengguna..... 23

4.3.2 Mendapatkan List Rekomendasi Lokasi..... 25

4.3.4 Mendapatkan Informasi Detail Objek WiFi 28

4.3.5 Perancangan Basis Data..... 30

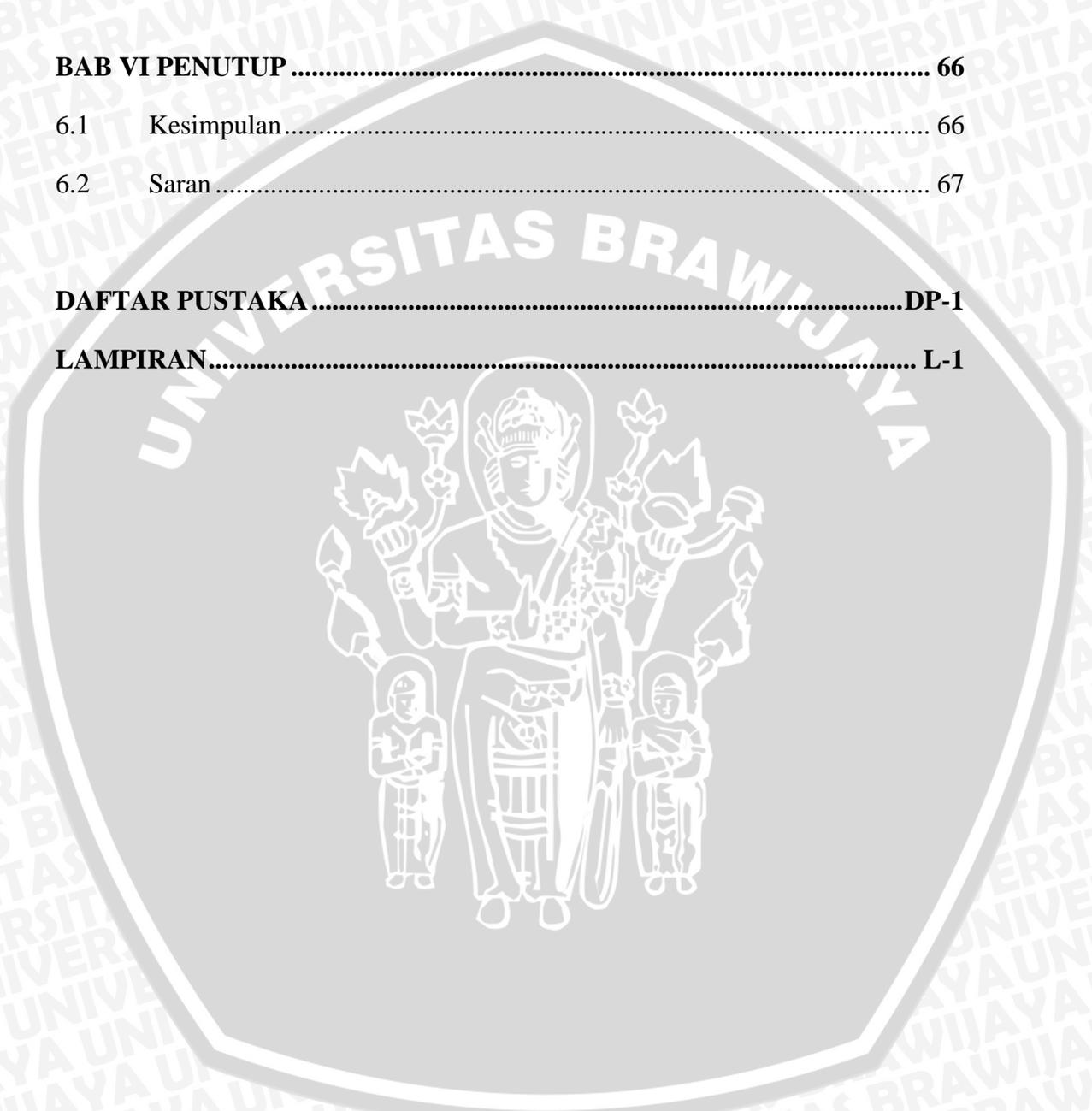
4.3.6 Perancangan Antarmuka..... 32

4.4 Proses Perhitungan 36



4.4.1	Perhitungan Jarak	36
4.4.2	Perhitungan Rekomendasi	37
BAB V IMPLEMENTASI DAN PENGUJIAN		40
5.1	Lingkungan Implementasi	40
5.1.1	Lingkungan Perangkat Keras.....	40
5.1.2	Lingkungan Perangkat Lunak.....	41
5.2	Batasan Implementasi.....	41
5.3	Implementasi Basis Data	42
5.4	Implementasi <i>Class</i>	42
5.5	Implementasi Kode Program	43
5.5.1	Kode Program Mendapatkan Lokasi Pengguna	43
5.5.2	Kode Program Mendapatkan List Rekomendasi Lokasi	44
5.5.3	Kode Program Mendapatkan Informasi Detail Objek WiFi.....	49
5.5.4	Kode Program JSON	50
5.6	Implementasi Antarmuka	52
5.6.1	Implementasi Antarmuka Halaman Utama	52
5.6.2	Implementasi Antarmuka Halaman Rekomendasi	53
5.6.3	Implementasi Antarmuka Halaman Detail	54
5.7	Pengujian	55
5.7.1	Pengujian Validasi Fitur	56
5.7.2	Pengujian Akurasi Sistem.....	58
5.7.3	Pengujian <i>Usability</i>	61
5.8	Hasil dan Analisis Pengujian.....	62
5.8.1	Analisis Hasil Pengujian Validasi Fitur.....	62

5.8.2	Analisis Hasil Pengujian Akurasi Sistem	62
5.8.3	Analisis Hasil Pengujian <i>Usability</i>	63
BAB VI PENUTUP		66
6.1	Kesimpulan	66
6.2	Saran	67
DAFTAR PUSTAKA		DP-1
LAMPIRAN		L-1



DAFTAR GAMBAR

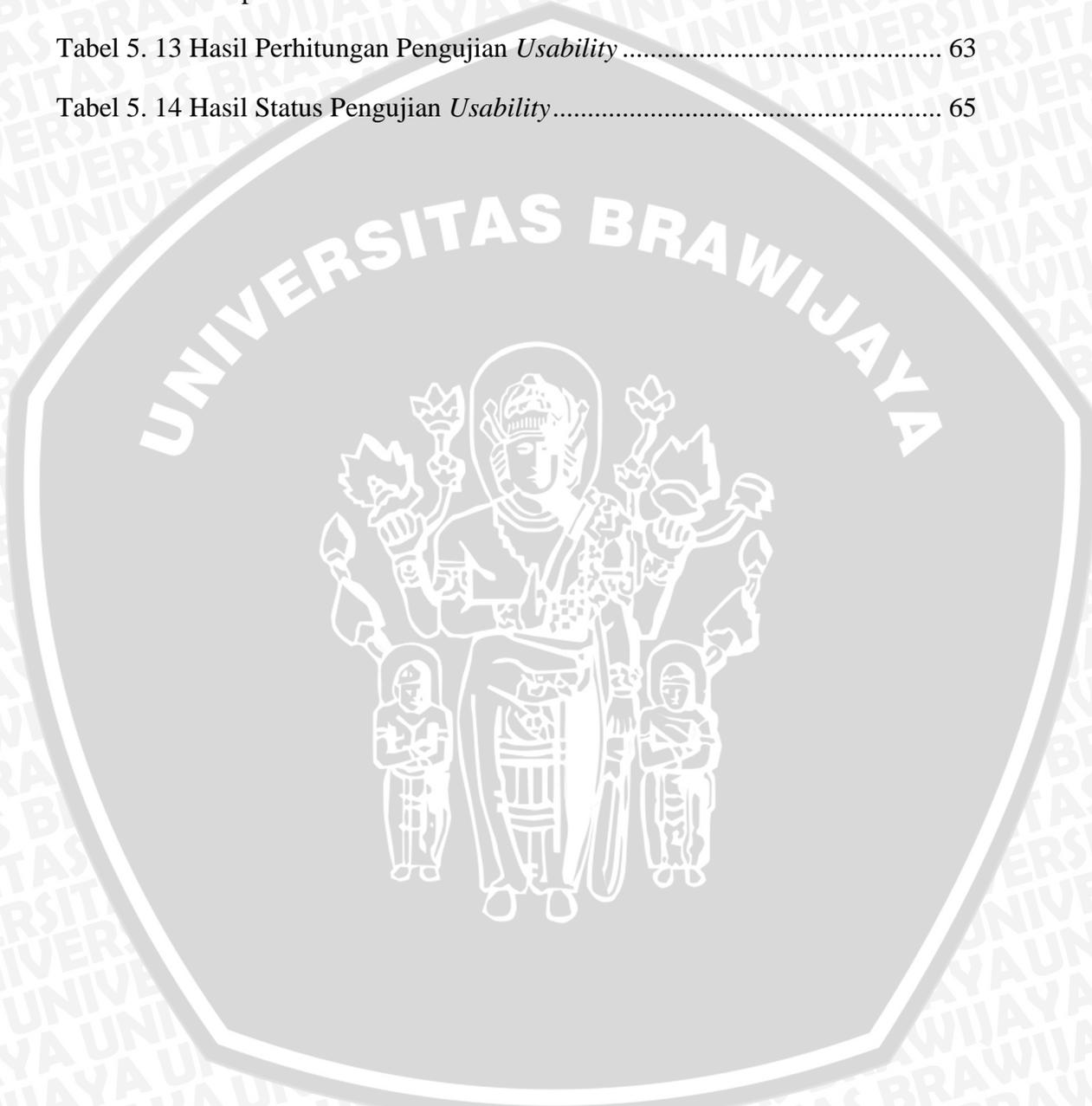
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	10
Gambar 3.2 Tampilan Aplikasi Speedtest pada Android.....	12
Gambar 4.1 Gambaran kerja sistem.....	17
Gambar 4.2 Diagram <i>Use case</i>	20
Gambar 4.3 <i>Activity Diagram</i> Mendapatkan Lokasi Pengguna.....	24
Gambar 4.4 <i>Sequence Diagram</i> Mendapatkan Lokasi Pengguna.....	24
Gambar 4.5 <i>Class Diagram</i> Mendapatkan Lokasi Pengguna.....	25
Gambar 4.6 <i>Activity Diagram</i> Mendapatkan List Rekomendasi Lokasi.....	26
Gambar 4.7 <i>Sequence Diagram</i> Mendapatkan List Rekomendasi Lokasi.....	27
Gambar 4.8 <i>Class Diagram</i> Mendapatkan List Rekomendasi Lokasi.....	28
Gambar 4.9 Mendapatkan Informasi Detail Objek Wifi.....	29
Gambar 4.10 <i>Sequence Diagram</i> Mendapatkan Informasi Detail Objek Wifi.....	29
Gambar 4.11 <i>Class Diagram</i> Mendapatkan Informasi Detail Objek Wifi.....	30
Gambar 4.12 Rancangan Alur Proses dan Antarmuka.....	32
Gambar 4.13 Rancangan Antarmuka Halaman Utama.....	33
Gambar 4.14 Antar Muka Halaman Rekomendasi.....	34
Gambar 4.15 Antarmuka Halaman Detail.....	35
Gambar 5.1 Implementasi Antarmuka Halaman Utama.....	53
Gambar 5.2 Implementasi Antarmuka Halaman Rekomendasi.....	54
Gambar 5.3 Antarmuka Halaman Detail.....	55

DAFTAR TABEL

Tabel 2.1 Contoh Penilaian Jawaban Kuantitatif.....	9
Tabel 3.1 Parameter Nilai Kenyamanan	12
Tabel 3.2 Data Prioritas Responden.....	13
Tabel 3.3 Angka Prioritas Kriteria	13
Tabel 4. 1 Spesifikasi Kebutuhan Fungsional.....	19
Tabel 4. 2 Skenario <i>Use case</i> Mendapatkan Lokasi Pengguna.....	20
Tabel 4. 3 Skenario <i>Use case</i> Mendapatkan List Rekomendasi Lokasi	21
Tabel 4. 4 Skenario <i>Use case</i> Mendapatkan Informasi Detail Objek WiFi.....	22
Tabel 4. 5 Spesifikasi Kebutuhan Non Fungsional.....	23
Tabel 4. 6 Struktur Tabel	30
Tabel 4. 7 Keterangan Desain Antarmuka Halaman Utama	33
Tabel 4. 8 Keterangan Desain Antarmuka Halaman Rekomendasi.....	34
Tabel 4. 9 Keterangan Desain Antarmuka Halaman Detail.....	35
Tabel 4.10 Tabel Nilai Alternatif	38
Tabel 5. 1 Lingkungan Perangkat Keras Pengembangan Aplikasi	40
Tabel 5. 2 Lingkungan Perangkat Keras Instalasi dan Pengujian Aplikasi	40
Tabel 5. 3 Lingkungan Perangkat lunak Pengembangan Aplikasi	41
Tabel 5. 4 Lingkungan Perangkat Lunak Instalasi dan Pengujian Aplikasi	41
Tabel 5. 5 Implementasi Perancangan <i>Class</i>	42
Tabel 5. 6 Kasus Uji Mendapatkan Lokasi Pengguna	56
Tabel 5. 7 Kasus Uji Mendapatkan List Rekomendasi Lokasi	56
Tabel 5. 8 Kasus Uji Mendapatkan Informasi Detail Objek WiFi.....	57
Tabel 5. 9 Hasil Pengujian Validasi.....	57



Tabel 5. 10 Hasil Pengujian Akurasi Sistem Terhadap Perhitungan Manual	59
Tabel 5. 11 Hasil Pengujian <i>Usability</i>	61
Tabel 5. 12 Interpretasi Skor Likert	63
Tabel 5. 13 Hasil Perhitungan Pengujian <i>Usability</i>	63
Tabel 5. 14 Hasil Status Pengujian <i>Usability</i>	65



BAB I PENDAHULUAN

1.1 Latar Belakang

Internet merupakan suatu hal yang sangat dibutuhkan pada jaman sekarang. Internet menjadi sebuah fasilitas yang memudahkan semua orang dalam melakukan berbagai kegiatan dan transaksi. Internet juga menjadi sebuah jembatan dalam menemukan informasi yang dibutuhkan oleh penggunanya. Pengguna Internet di Indonesia mencapai 16.72% [ILA-14] dengan pengguna melalui *mobile/smartphone* mencapai 14% dari total penduduk [BMC-14] dan banyak diantaranya menyandang status sebagai mahasiswa.

Banyak cara yang fleksibel bagi mahasiswa untuk tersambung ke jaringan Internet. Salah satunya dengan menggunakan fasilitas Internet yang tersedia di tempat umum seperti universitas, café, dan lain-lain. Berdasarkan hasil survei yang dilakukan sebelum penelitian ini dilaksanakan, dari 100 mahasiswa yang pernah/sedang berkuliah di Malang sebanyak 42% menggunakan fasilitas Internet yang tersedia di tempat umum karena mencari jaringan Internet yang stabil dan kencang, 31% merasa paket Internet terlalu mahal dan ingin menghemat kuota serta mencari yang gratis, 21% memaksimalkan fasilitas sambil melakukan kegiatan lain serta mencari suasana, dan 6% tidak memiliki Internet di tempat tinggal.

Kebutuhan mahasiswa yang menggunakan fasilitas Internet di tempat umum pun cukup beragam. Sebanyak 76% mahasiswa menggunakan fasilitas tersebut untuk *browsing* dan diselingi dengan kegiatan *download*, *streaming video*, serta bermain *game online* dan 17% sengaja menggunakannya untuk mengunduh sesuatu. Sedangkan 7% lainnya menggunakannya hanya untuk *streaming video*. Namun tidak semua tempat umum dengan koneksi Internet menjadi pilihan para mahasiswa. Dari 100 responden, sebanyak 71% responden menjadikan kecepatan Internet pada prioritas yang paling utama. Mahasiswa yang

menjadikan faktor kenyamanan lokasi sebagai prioritas utamanya sebanyak 23%. Sedangkan 6% sisanya memilih biaya dan jarak sebagai prioritas yang utama.

Smartphone Android memiliki fitur *Global Positioning System* (GPS) yang dapat digunakan untuk memberikan informasi geolokasi terkini pengguna. *Platform* Android juga dapat diintegrasikan dengan Google Maps API v2 ataupun Maps With Me API yang menawarkan kemudahan dalam membangun aplikasi yang memerlukan data geolokasi. Selain itu, *smartphone* Android merupakan salah satu device dengan harga terjangkau dan populer dengan lebih dari 800 juta pengguna sejak diperkenalkan tahun 2008 [WAU-13].

Optimalisasi aplikasi perangkat gerak memiliki 3 ide utama yaitu mengurangi fitur, konten, serta memperbesar tampilan elemen. Pengurangan fitur dilakukan untuk menghilangkan sesuatu yang tidak penting pada aplikasi perangkat gerak. Pengurangan konten berguna untuk mengurangi kata serta mengatur informasi sekunder. Memperbesar tampilan tiap elemen berfungsi untuk mengakomodasi pengguna yang memiliki jari yang besar [NNG-12]. Sehingga diperlukan pengujian *usability* pada aplikasi yang diimplementasikan pada perangkat gerak.

weighted product merupakan salah satu metode perhitungan yang dapat diimplementasikan dalam sistem rekomendasi. Penelitian "*Simulation Modeling of Tablet PCs Selection Using Weighted-Product Algorithm*" merupakan penelitian yang menerapkan metode *weighted product*. Penelitian tersebut membuktikan bahwa metode *weighted product* dapat disimulasikan dalam sistem sebagai sistem pengambil keputusan dalam pemilihan Tablet PC untuk konsumen [SMT-14]. Beberapa rumus seperti Haversine atau *Euclidian* dapat digunakan untuk melakukan perhitungan jarak berdasarkan geolokasi pengguna dan lokasi yang ada.

Oleh karena jarak lokasi pengguna dapat dikalkulasi secara otomatis, maka dimungkinkan untuk dibuatnya sebuah aplikasi pada perangkat bergerak yang dapat memberikan rekomendasi mengenai lokasi jaringan Internet di tempat umum berdasarkan faktor jarak lokasi dari pengguna, kecepatan Internet, biaya

yang harus dikeluarkan, dan tingkat kenyamanan tempat. Sehingga, diharapkan aplikasi dapat membantu para pengguna aplikasi untuk menemukan dan memilih tempat-tempat yang menyediakan jaringan Internet yang sesuai, lengkap dengan informasi tambahan yang berupa ulasan singkat terkait tempat yang menyediakan layanan Internet tersebut.

1.2 Rumusan Masalah

Dari latar belakang diatas, permasalahan yang akan dibahas pada penelitian ini adalah sebagai berikut:

1. Bagaimana membangun dan mengimplementasikan aplikasi yang bisa memberikan rekomendasi lokasi jaringan Internet di tempat umum berdasarkan jarak antara pengguna dengan tempat yang menyediakan layanan internet dengan pengguna saat itu, kecepatan Internet, biaya yang dikeluarkan, dan tingkat kenyamanan lokasi pada perangkat bergerak berbasis Android?
2. Seberapa besar tingkat akurasi penerapan metode *weighted product* terhadap perhitungan yang dilakukan secara manual jika diimplementasikan pada sebuah program aplikasi pada perangkat bergerak berbasis Android yang dapat memberikan rekomendasi terkait lokasi penyedia layanan Internet dengan melibatkan faktor jarak lokasi, kecepatan Internet, biaya yang dikeluarkan, serta kenyamanan tempat tersebut?
3. Bagaimana efektivitas aplikasi yang dibangun dalam memberikan rekomendasi?
4. Apakah aplikasi yang dibangun dalam perangkat bergerak berbasis Android dapat memberikan nyaman dan kemudahan dalam penggunaanya?

1.3 Batasan Masalah

Dalam penelitian ini, permasalahan akan dibatasi sebagai berikut:

1. Aplikasi yang akan dibangun hanya dapat digunakan pada sistem operasi Android

2. Aplikasi ini hanya menampilkan sebagian data atau sampel dari café yang memiliki jaringan Internet di kota Malang namun bisa dikembangkan dengan menambah tempat dan kota.
3. Penelitian dititik beratkan pada rancang bangun aplikasi.
4. Aktor yang berinteraksi pada aplikasi hanya pengguna aplikasi.

1.4 Tujuan Penelitian

Tujuan dari dilakukanya penelitian ini yaitu:

1. Membuat sebuah aplikasi pada perangkat bergerak Android yang dapat memberikan rekomendasi terkait lokasi jaringan Internet di tempat umum berdasarkan jarak, kecepatan Internet, biaya yang dikeluarkan, dan kenyamanan pada Android yang sesuai dengan pengguna.
2. Menguji akurasi metode *weighted product* yang diimplementasikan pada sebuah aplikasi berbasis Android dengan hasil rekomendasi yang dihasilkan dari perhitungan yang dihitung secara manual.
3. Melakukan pengujian untuk mengetahui efektivitas aplikasi yang dibangun.
4. Menguji *usability* aplikasi untuk mengetahui tingkat kenyamanan dan kemudahan pada penggunanya.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari pembangunan perangkat lunak ini adalah menyediakan aplikasi *mobile* yang memberikan rekomendasi untuk menemukan lokasi jaringan Internet di tempat umum beserta informasi tambahan yang dibutuhkan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang literatur yang digunakan terkait penelitian dan pembangunan aplikasi. Bab ini menjelaskan tentang rumus Haversine, metode *weighted product*, AsyncTask, dan *usability testing*.

2.1 Rumus Haversine

Rumus Haversine merupakan persamaan pada navigasi yang memberikan jarak lingkaran besar antara dua titik bumi berdasarkan *longitude* dan *latitude*. Rumus ini diperkenalkan oleh Prof. James Inman pada tahun 1835. Rumus Haversine merupakan perhitungan yang cukup akurat dimana rumus ini mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi [ADC-14].

Rumus Haversine untuk perhitungan jarak ditunjukkan pada rumus 2-1:

$$d=2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos\phi_1 \cos\phi_2 \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right) \quad (2-1)$$

Dimana :

d : jarak

r : 6371.1 (konstanta radius bumi dalam satuan km)

$\Delta\phi$: Selisih *Latitude*

$\Delta\lambda$: Selisih *Longitude*

ϕ_1 : *Latitude* awal

ϕ_2 : *Latitude* akhir

2.2 Metode *Weighted Product*

weighted product merupakan salah satu metode yang ditawarkan untuk menyelesaikan masalah dalam *Multi Attribute Decision Making* (MADM). Metode ini disebut sebagai analisis berdimensi karena struktur matematikanya menghilangkan satuan ukuran [FMA-06]. Metode ini juga merupakan salah satu metode dalam MADM dengan proses komputasi yang sederhana [MAD-95].

Metode *weighted product* menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot

yang bersangkutan. Hasil dari perkalian tersebut kemudian menghasilkan preferensi alternatif yang dianalogikan sebagai vektor S. Perhitungan preferensi alternatif ditunjukkan pada rumus 2-2:

$$S_i = \prod_{j=1}^n X_{ij}^{W_j} \quad (2-2)$$

Dimana :

S : Preferensi alternatif dianalogikan sebagai vektor S

X : Nilai kriteria

W : Bobot Kriteria

i : Alternatif

j : kriteria

n : banyaknya kriteria

Bobot kriteria yang ada disimbolkan sebagai W_j dimana $\sum W_j = 1$. W_j merupakan pangkat bernilai positif untuk attribute manfaat dan bernilai negatif untuk atribut biaya. Perhitungan bobot kriteria ditunjukkan pada rumus 2-3:

$$W_j = \frac{w_j}{\sum w_j} \quad (2-3)$$

Diamana :

W : bobot kriteria

j : kriteria

Hasil dari vektor S kemudian dijumlahkan sehingga menjadi pembagi untuk menghitung preferensi untuk perankingan yang dianalogikan sebagai vektor V. Hasil dari vektor V didapatkan dari pembagian antara vektor S dan hasil dari penjumlahan vektor S. Perhitungan preferensi untuk perankingan ditunjukkan pada rumus 2-4:

$$V = \frac{\prod_{j=1}^n X_{ij}^{W_j}}{\prod_{j=1}^n (X_j^{W_j})^{w_j}} \quad (2-4)$$

Dimana :

V : preferensi untuk perankingan dianalogikan sebagai vektor V

X : nilai kriteria

W : Bobot kriteria/subkriteria

i : Alternatif

j : kriteria

n : Banyaknya kriteria

Hasil dari vektor V kemudian diurutkan dari nilai tertinggi ke nilai terendah sehingga menghasilkan nilai perankingan untuk rekomendasi.

2.3 AsyncTask

AsyncTask merupakan sebuah kelas abstrak yang disediakan untuk Android. AsyncTask membantu pengembang untuk menggunakan UI thread dengan tepat. Kelas ini akan langsung menampilkan operasi di *background* thread tanpa harus memanipulasi *thread*. [CTE-13]

AsyncTask memiliki 4 metode yaitu :

- doInBackground

Metode ini menjalankan komputasi pada *background* yang bisa memakan banyak waktu. Hasil komputasi pada metode ini akan menghasilkan kembalian dan akan dilanjutkan ke tahap akhir.

- onPostExecute

Hasil kembalian dari metode doInBackground akan diproses pada metode ini sebagai parameter. Metode ini akan dipanggil di UI *Thread* setelah komputasi pada *background* selesai.

- onPreExecute

Metode ini akan dipanggil sebelum doInBackground dieksekusi. Metode ini biasanya digunakan untuk menampilkan *progress bar* pada UI.

- onProgressUpdate

Metode ini akan dipanggil dengan ketika metode doInBackground memanggil publishProgress.

2.4 Usability Testing

Usability testing merupakan sebuah pengujian yang mengkaji seberapa mudah aplikasi digunakan. Kata “*usability*” mengacu pada metode untuk melakukan improvisasi terhadap ease-of use selama proses desain. [CMS-10]

Para ahli sepakat bahwa *usability* testing mencakup 5 komponen yaitu :

1. *Learnability*

Semudah apa pengguna dapat mempelajari penggunaan produk tersebut untuk pertama kali.

2. *Efficiency*

Secepat apa pengguna dapat melakukan tugasnya.

3. *Memorability*

Sejauh mana pengguna dapat mengingat langkah-langkah atau proses yang dilakukan untuk mencapai tujuannya.

4. *Errors*

Sebanyak apa, sejauh mana, dan semudah apa bagi pengguna jika berhubungan pada eror yang ada.

5. *Satisfaction*

Bagaimana tanggapan pengguna terhadap desain produk secara keseluruhan.

2.4.1 Kuesioner USE

Kuesioner USE merupakan media kuesioner untuk mengukur *usability* dengan memakai 3 parameter yaitu kegunaan (*usefulness*), kepuasan (*satisfaction*), dan *ease of use*. *Ease of use* merupakan sebuah parameter yang dibagi menjadi 2 faktor yaitu kemudahan dalam penggunaan (*ease of use*) dan kemudahan dalam mempelajari aplikasi (*ease of learning*) [STC-01].

Contoh *item* dari pertanyaan dalam kuesioner USE adalah sebagai berikut:

- *Usefulness*
 - Aplikasi ini dalam pengerjaannya memenuhi ekspektasi saya.
 - Aplikasi ini membuat saya menjadi lebih produktif.
 - Aplikasi ini sangat berguna.
- *Ease of Use*
 - Aplikasi ini mudah digunakan.
 - Aplikasi ini *user-friendly*.
 - Aplikasi ini fleksibel.
- *Ease of Learning*
 - Saya belajar dengan cepat dalam penggunaannya.

- Aplikasi ini mudah diingat dalam penggunaannya.
- Aplikasi ini dapat dengan mudah saya pelajari.
- *Satisfaction*
 - Saya puas dengan aplikasi ini.
 - Aplikasi ini menyenangkan untuk digunakan.
 - Saya merasa saya harus memiliki aplikasi ini.

2.5 Skala Likert

Skala Likert merupakan skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Nilai dari skala Likert tergantung kebutuhan. [ADR-10]

Skala Likert menjabarkan variabel yang diukur menjadi indikator variabel. Indikator tersebut kemudian dijadikan sebagai titik tolak untuk menyusun *item-item* instrumen yang dapat berupa pernyataan atau pertanyaan.

Jawaban setiap pertanyaan atau pernyataan yang menggunakan skala Likert mempunyai gradasi dari sangat positif sampai sangat negatif yang dapat berupa kata-kata sebagai contoh:

- a. Sangat setuju
- b. Setuju
- c. Ragu-ragu
- d. Tidak setuju
- e. Sangat tidak setuju

Untuk keperluan analisis kuantitatif, maka jawaban itu dapat diberi skor sebagai contoh pada Tabel 2.1.

Tabel 2.1 Contoh Penilaian Jawaban Kuantitatif

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

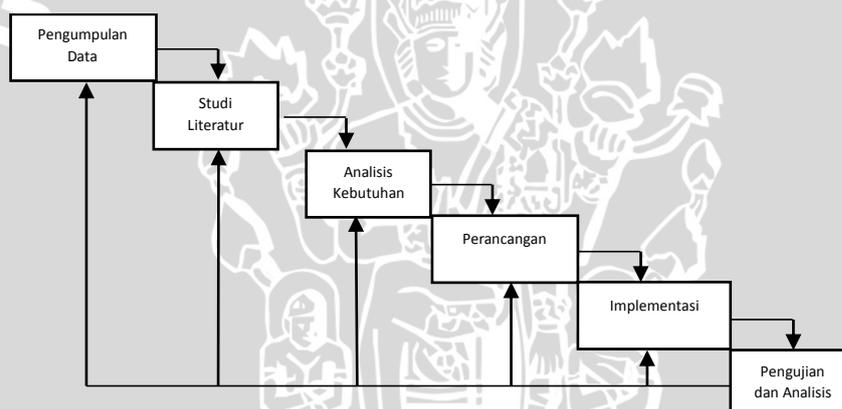
BAB III

METODOLOGI PENELITIAN

Bab ini berisi tentang metode penelitian yang akan dilakukan dalam perancangan dan pembangunan aplikasi rekomendasi lokasi WiFi *hotspot* di tempat umum bagi mahasiswa. Metodologi penelitian akan menjelaskan tentang rincian alur yang akan dilakukan dalam penelitian.

3.1 Metodologi Penelitian

Metode penelitian yang dilakukan mencakup studi literatur, pengumpulan data, analisis kebutuhan, perancangan, implementasi, dan pengujian. Diagram alir metodologi penelitian terdapat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1.1 Pengumpulan Data

Penelitian membutuhkan data yang berkaitan dengan penilaian responden terhadap jarak, biaya yang dikeluarkan, kecepatan Internet, dan parameter kenyamanan. Keempat parameter yang ada didapatkan dari hasil wawancara non formal. Data dikumpulkan dari 100 mahasiswa di kota Malang yang kemudian menghasilkan bobot nilai dan parameter skor kenyamanan. Bobot nilai kemudian digunakan untuk perhitungan yang menghasilkan rekomendasi lokasi. Selain itu penelitian melibatkan 30 mahasiswa sebagai penguji pada uji *usability*.

Penelitian juga membutuhkan data sampel yang berisi letak lokasi, biaya minimum yang harus dikeluarkan, dan kecepatan Internet yang nantinya digunakan pada aplikasi yang dibangun. Sampel terdiri dari 10 kafe yang ada di kota Malang. Sampel ini digunakan untuk pengujian yang akan dilakukan setelah tahap perancangan. Kecepatan Internet dari sampel didapatkan melalui aplikasi Android bernama Speedtest.

3.1.1.1 Pengambilan Nilai Kriteria

Kriteria yang digunakan dalam penelitian ini meliputi 4 hal yaitu jarak, biaya yang dikeluarkan, nilai kenyamanan, dan kecepatan Internet. Nilai untuk kriteria biaya, kenyamanan, dan kecepatan Internet yang ada diambil langsung oleh peneliti melalui survei lapangan dengan mendatangi lokasi yang akan dijadikan sampel. Selain itu, peneliti juga mengambil nilai *longitude* dan *latitude* lokasi yang didapatkan dari Google Maps.

Peneliti sebelumnya telah melakukan penelitian dengan cara menyebarkan angket yang berkaitan dengan prioritas mahasiswa dalam memilih tempat pada 100 mahasiswa.

Pengambilan nilai kriteria yang ada adalah sebagai berikut:

- Biaya

Biaya yang menjadi nilai kriteria diambil dari biaya minimal yang ada pada menu atau biaya minimum pembelian untuk mendapatkan akses WiFi. Biaya yang diambil merupakan survei lapangan penulis terhadap lokasi-lokasi yang memiliki akses WiFi di kota Malang.

- Nilai Kenyamanan

Parameter nilai kenyamanan merupakan hasil perhitungan dari penelitian yang telah dilakukan sebelumnya. Parameter tersebut berupa tersedianya stop kontak, tata ruang yang luas, kebersihan, adanya kamar mandi, adanya AC, adanya music/live music, keramahan pelayanannya, adanya smoking room, dan adanya tempat ibadah. Nilai kenyamanan yang ada diambil dari jumlah responden yang memilih parameter kenyamanan pada kuesioner diberikan di awal penelitian. Nilai kenyamanan untuk parameter

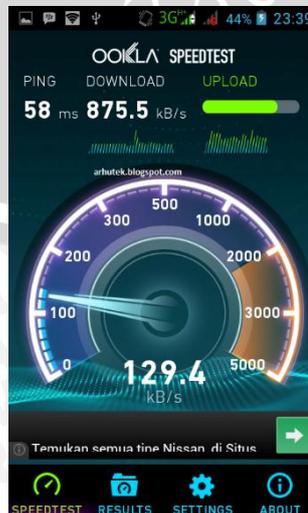
tersebut diambil dari jumlah pilihan kriteria oleh responden. Nilai kenyamanan terdapat pada Tabel 3.1.

Tabel 3.1 Parameter Nilai Kenyamanan

Kriteria	Nilai
Adanya Stop Kontak	69
Kebersihan	53
Luasnya Tata Ruang	60
Adanya Kamar Mandi	42
Adanya AC	29
Adanya Musik/Live Music	19
Keramahan Pelayanan	26
Adanya Smoking Room	27
Adanya Tempat Ibadah	34
Total Nilai	359

- Kecepatan Internet

Nilai dari kecepatan Internet diambil dari hasil *download* oleh aplikasi Speedtest.net pada Android dengan besaran *kilobyte per second* (kBps). Speedtest merupakan layanan yang banyak digunakan untuk melakukan uji kecepatan *bandwith* Internet. Layanan ini dikembangkan oleh Ookla Net Metrics yang diluncurkan pada tahun 2006 [ANC-12]. Tampilan dari aplikasi Speedtest pada Android ditunjukkan dalam Gambar 3.2.



Gambar 3.2 Tampilan Aplikasi Speedtest pada Android

3.1.1.2 Pengambilan Bobot Kriteria

Bobot dari masing-masing kriteria tersebut diambil dari hasil perhitungan pada penelitian awal. Pembobotan tersebut diambil dari jumlah prioritas pertama hingga prioritas keempat dari responden terhadap tiap kriteria. Data prioritas responden ditunjukkan pada Tabel 3.2:

Tabel 3.2 Data Prioritas Responden

	P1	P2	P3
Biaya	5	23	32
Jarak	1	14	39
Kecepatan Internet	71	20	9
Kenyamanan	23	42	19
Jumlah Responden	100	99	99

Dari data prioritas diatas kemudian didapatkan angka prioritas tiap kriteria sebagai berikut:

Tabel 3.3 Angka Prioritas Kriteria

Kecepatan Internet	Kenyamanan	Biaya	Jarak
71	42	32	39

Angka diatas kemudian dihitung berdasarkan rumus (2-3) sehingga didapatkan hasil sebagai berikut:

Perhitungan pembobotan:

$$\text{Pembagi} = 71+42+32+39=184$$

$$\text{Bobot Kecepatan Internet} = \frac{71}{184}=0.39$$

$$\text{Bobot Kenyamanan} = \frac{42}{184}=0.23$$

$$\text{Bobot Biaya} = \frac{32}{184}=0.17$$

$$\text{Bobot Jarak} = \frac{39}{184}=0.21$$

Dari hasil perhitungan diatas maka didapatkan bobot kriteria untuk kecepatan Internet sebesar 0.39 dan bobot kriteria untuk kenyamanan sebesar

0.23. Bobot kriteria untuk biaya sebesar 0.17 dan bobot kriteria untuk jarak sebesar 0.21. Bobot kriteria jarak dan biaya bernilai negatif karena kedua bobot tersebut merupakan atribut biaya.

3.1.2 Studi Literatur

Tahap ini merupakan tahap pengumpulan pengetahuan dari banyak sumber seperti buku, dokumen, dan artikel yang diambil dari berbagai sumber. Sumber pengetahuan tersebut dijadikan sebagai landasan teori untuk merancang aplikasi rekomendasi lokasi. Pengetahuan tersebut meliputi :

- Rumus Haversine
Bagian ini menjelaskan tentang proses perhitungan jarak antar 2 koordinat menggunakan rumus Haversine
- *weighted product*
Bagian ini menjelaskan tentang proses perhitungan skor rekomendasi menggunakan *weighted product*
- AsyncTask
Bagian ini menjelaskan tentang pengertian dan implementasi AsyncTask di Android.
- Pengujian *Usability*
Bagian ini digunakan sebagai acuan dalam melakukan tahap pengujian terhadap aplikasi yang akan dibangun. Bagian ini menjelaskan tentang tujuan dan cara pengujian yang akan digunakan.

3.1.3 Analisis Kebutuhan

Proses analisis kebutuhan diawali dari deskripsi aplikasi serta daftar kebutuhan yang ada untuk perancangan aplikasi. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem dan rincian dari alur aktifitas serta pengguna. Kebutuhan pengguna didapatkan dari hasil kesimpulan pengumpulan data yang dilakukan sebelumnya. Kebutuhan yang diperlukan aplikasi terdapat pada kebutuhan fungsional.

3.1.4 Perancangan

Perancangan dilakukan setelah semua kebutuhan sistem didapatkan dari melalui tahap analisis kebutuhan. Semua kebutuhan yang ada akan dimodelkan dengan UML (*Unified Modeling Language*).

3.1.4.1 Perancangan Fungsional

Perancangan pada tahap ini bertujuan untuk menggambarkan fungsi yang ada pada aplikasi dalam bentuk *use case*. Pada tahap ini terdapat aksi mana saja yang bisa dilakukan oleh aktor yang telah ditetapkan pada tahap analisis kebutuhan. Setiap aksi kemudian dijabarkan pada tabel skenario yang kemudian menjadi dasar pembuatan *activity diagram*.

3.1.4.2 Perancangan Aktifitas

Perancangan aktifitas antara pengguna dan aplikasi dimodelkan dalam bentuk *activity diagram*. *Activity diagram* yang ada didapatkan dari *basic flow* yang terdapat pada tabel skenario yang telah dibuat sebelumnya.

3.1.4.3 Perancangan Interaksi

Perancangan interaksi antar objek dalam sistem yang telah teridentifikasi dimodelkan dalam *sequence diagram*. Diagram ini berupa skenario atau rangkaian langkah-langkah yang dilakukan dari sebuah kegiatan untuk menghasilkan output tertentu. *Sequence diagram* disusun berdasarkan *activity diagram* yang dibuat sebelumnya.

3.1.4.4 Perancangan Kelas

Tahap ini dilakukan setelah perancangan fungsi. Perancangan ini memodelkan sistem kedalam bentuk *class diagram*. Diagram ini digunakan untuk mengidentifikasi kelas-kelas serta yang terdapat dalam sistem. *Class diagram* disusun berdasarkan *sequence diagram* yang dibuat sebelumnya.

3.1.4.5 Perancangan Basis Data

Perancangan ini berisi tabel dan kolom apa saja yang dibutuhkan untuk membangun aplikasi rekomendasi lokasi WiFi. Perancangan ini juga berisi

tentang contoh isi tiap kolom dan kegunaanya. Tabel dan kolom yang dibutuhkan dalam perancangan basis data ditetapkan setelah tahap analisis kebutuhan.

3.1.4.6 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk mendesain tampilan yang sesuai pada perangkat berbasis Android. Perancangan antarmuka yang dilakukan berdasarkan perancangan fungsional sistem yang ada.

3.1.5 Implementasi

Implementasi dilakukan setelah melalui fase perancangan. Implementasi menerapkan alur proses aplikasi yang sudah didapat dan disatukan dengan desain tampilan aplikasi. Implementasi dilakukan dengan membuat kode program yang langsung di-*compile* menggunakan perangkat bergerak bersistem operasi Android. Hal ini dilakukan untuk mengurangi kendala dalam menjalankan aplikasi. Program yang digunakan untuk membangun aplikasi ini adalah Eclipse.

3.1.6 Pengujian dan Analisis

Pengujian dilakukan untuk menunjukkan agar perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan dasar. Pengujian dilakukan menggunakan metode *blackbox testing*. *Blackbox testing* digunakan untuk mengetahui keseluruhan fungsionalitas dari aplikasi yang telah dibangun secara efektif dan efisien.

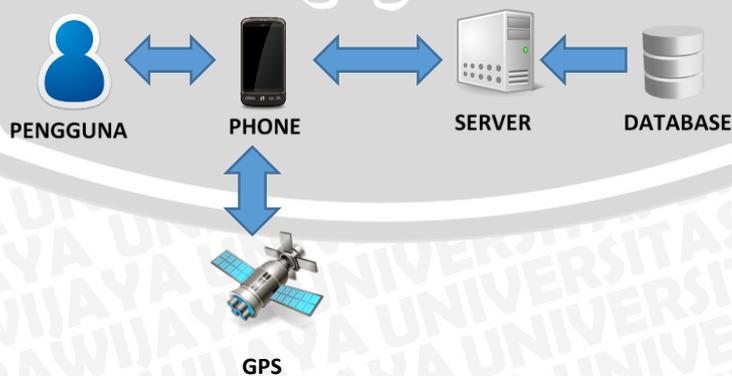
Strategi pengujian yang digunakan adalah *validation testing* dan uji akurasi sistem dimana aplikasi akan diuji berdasarkan kecocokan fungsi yang berjalan dengan perancangan yang ada serta perhitungan manual dan perhitungan oleh sistem. Pengujian *usability* juga dilakukan untuk mengetahui nilai kegunaan aplikasi untuk pengguna aplikasi nantinya. Analisis kemudian dilakukan untuk mendapatkan dari hasil dan kesimpulan.

BAB IV PERANCANGAN

Bab ini membahas tentang perancangan aplikasi rekomendasi lokasi WiFi pada *smartphone* Android. Perancangan dilakukan dalam 3 tahap yaitu perancangan umum sistem, analisis kebutuhan, dan perancangan perangkat lunak. Perancangan umum sistem merupakan penjelasan tentang gambaran umum aplikasi rekomendasi lokasi WiFi. Tahap analisis kebutuhan berisi tentang analisis aktor, deskripsi kebutuhan fungsional dan diagram *use case*. Tahap perancangan perangkat lunak meliputi perancangan *activity diagram*, pemodelan *sequence diagram*, pemodelan *class diagram*, perancangan basis data dan perancangan antarmuka pada *smartphone* Android.

4.1 Deskripsi Umum Aplikasi

Aplikasi rekomendasi lokasi WiFi di tempat umum merupakan aplikasi yang dibangun untuk sistem operasi Android. Aplikasi ini memberikan rekomendasi lokasi WiFi berdasarkan jarak dari tempat berdiri, biaya yang dikeluarkan, kecepatan Internet, dan kenyamanan dari daftar lokasi yang ada dalam *database*. Pengguna cukup menekan tombol rekomendasi untuk mendapatkan rekomendasi sesuai perhitungan yang dilakukan oleh sistem. Gambaran kerja sistem digambarkan pada Gambar 4.1.



Gambar 4.1 Gambaran kerja sistem

Gambaran kerja sistem yang terdapat dalam Gambar 4.1 secara umum adalah sebagai berikut :

1. Sistem melakukan set posisi pengguna melalui GPS.
2. Sistem meminta data nama, lokasi, skor kenyamanan, kecepatan Internet, dan biaya minimal setiap objek lokasi WiFi yang telah tersimpan di *database server*.
3. Server mengirimkan *data* yang diminta oleh sistem.
4. Sistem menghitung jarak antara lokasi pengguna dan lokasi WiFi dengan rumus haversine.
5. Sistem menghitung rekomendasi berdasarkan jarak, biaya, kecepatan Internet, dan skor kenyamanan yang ada dengan metode *weighted product*.
6. Pengguna mendapatkan hasil rekomendasi sistem
7. Pengguna bisa mendapatkan informasi detail lokasi yang menjadi rekomendasi

4.2 Analisis Kebutuhan

Aplikasi perangkat bergerak ini dikembangkan pada *platform* Android. Kegunaan dari aplikasi ini adalah memberikan rekomendasi lokasi WiFi *hotspot* yang ada di tempat umum pada pengguna.

4.2.1 Analisis Aktor

Aplikasi ini melibatkan aktor dalam penggunaannya, yaitu pengguna. Pengguna merupakan aktor yang bisa mengetahui lokasi pribadi, mendapatkan rekomendasi tempat yang dimaksud, serta mendapatkan informasi atas tempat yang direkomendasikan.

4.2.2 Kebutuhan Fungsional

Daftar kebutuhan berisi semua kebutuhan yang diperlukan oleh aplikasi yang akan dibangun. Kebutuhan fungsional merupakan kebutuhan yang terdiri dari fitur-fitur yang disediakan oleh aplikasi untuk memenuhi kebutuhan pengguna.

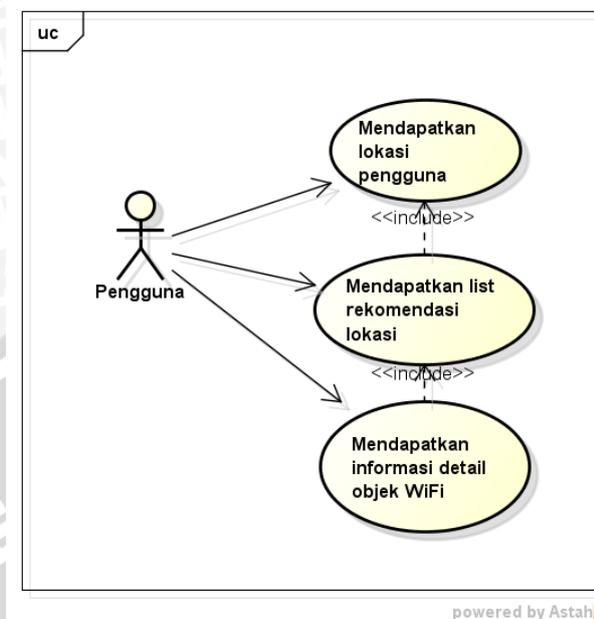
Kebutuhan fungsional ditunjukkan dengan penomoran SRS (*Software Requirement Specification*) serta digambarkan dengan menggunakan diagram *use case*. Daftar kebutuhan fungsional ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Spesifikasi Kebutuhan Fungsional

Nomor SRS	Kebutuhan	Use case
SRS_01	Aplikasi harus dapat mendapatkan dan menampilkan posisi pengguna pada <i>Google Maps</i> .	Mendapatkan Lokasi Pengguna
SRS_02	Aplikasi harus dapat memberikan rekomendasi berdasarkan jarak dari tempat berdiri, skor kenyamanan, kecepatan Internet, dan harga minimal lokasi yang dimaksud	Mendapatkan List Rekomendasi Lokasi
SRS_03	Aplikasi harus dapat menampilkan informasi tempat yang menjadi rekomendasi berupa alamat, fasilitas yang ada, kecepatan Internet rata-rata, dan ulasan tempat tersebut secara singkat	Mendapatkan Informasi Detail Ojek WiFi

4.2.3 Diagram *Use case*

Berdasarkan kebutuhan fungsional yang telah didapatkan sebelumnya, maka dapat digambarkan susunan *use case* seperti Gambar 4.2.



Gambar 4. 2 Diagram *Use case*

Skenario *use case* digunakan untuk menjelaskan secara lebih mendetail tentang masing-masing kebutuhan fungsional yang terdapat pada diagram *use case* pada Gambar 4.2.

Kebutuhan fungsional untuk mendapatkan lokasi pengguna dilakukan secara otomatis dilakukan oleh sistem. Sistem kemudian menampilkan lokasi pengguna di halaman utama. Tabel 4.2 merupakan skenario *use case* mendapatkan lokasi pengguna.

Tabel 4. 2 Skenario *Use case* Mendapatkan Lokasi Pengguna

Skenario <i>Use case</i> Mendapatkan Lokasi Pengguna	
Kode SRS	SRS_01
Nama <i>Use case</i>	Mendapatkan Lokasi Pengguna
Tujuan	Untuk mendapatkan dan menampilkan lokasi pengguna
Deskripsi	<i>Use case</i> ini digunakan untuk mendapatkan lokasi pengguna yang dilakukan oleh sistem kemudian menampilkannya di

	halaman utama
Aktor	Pengguna
Pemicu	Pengguna membuka aplikasi
Kondisi Awal	Pengguna akan membuka aplikasi
Skenario Utama (Basic Flow)	
<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Sistem mendapatkan lokasi pengguna melalui GPS atau paket data Internet 	
Kondisi Akhir	Sistem menampilkan lokasi pengguna

Longitude dan *latitude* pengguna kemudian dihitung dengan *longitude* dan *latitude* semua lokasi WiFi di tempat umum yang terdaftar di *database* untuk mendapatkan jarak. Jarak, kecepatan Internet, skor kenyamanan, dan harga diolah oleh sistem sehingga menghasilkan rekomendasi. Rekomendasi berupa *list* dengan urutan rekomendasi dari nilai terbesar. Tabel 4.3 merupakan skenario *use case* mendapatkan list rekomendasi.

Tabel 4.3 Skenario *Use case* Mendapatkan List Rekomendasi Lokasi

Skenario <i>Use case</i> Mendapatkan List Rekomendasi Lokasi	
Kode SRS	SRS_02
Nama <i>Use case</i>	Mendapatkan List Rekomendasi Lokasi
Tujuan	Untuk mendapatkan list rekomendasi lokasi
Deskripsi	<i>Use case</i> ini digunakan untuk mendapatkan list rekomendasi lokasi yang terdaftar dalam <i>database</i>
Aktor	Pengguna
Pemicu	Pengguna menekan tombol untuk mendapatkan rekomendasi

Kondisi Awal	Sistem menampilkan halaman utama aplikasi
Skenario Utama (Basic Flow)	
<ol style="list-style-type: none"> 1. Pengguna menekan tombol untuk mendapatkan rekomendasi 2. Sistem menghitung jarak berdasarkan <i>longitude</i> dan <i>latitude</i> dari pengguna dan semua lokasi yang terdaftar di <i>database</i>. 3. Sistem menghitung rekomendasi dengan mengolah jarak, kecepatan Internet, skor kenyamanan, dan biaya yang dikeluarkan 	
Kondisi Akhir	Sistem menampilkan rekomendasi berupa list dengan urutan nilai rekomendasi dari yang terbesar

List dari hasil rekomendasi bisa dipilih oleh pengguna untuk mendapatkan informasi lokasi. Tabel 4.4 skenario *use case* mendapatkan informasi detail objek WiFi.

Tabel 4. 4 Skenario *Use case* Mendapatkan Informasi Detail Objek WiFi

Skenario Use case Mendapatkan Informasi Detail Objek WiFi	
Kode SRS	SRS_03
Nama Use case	Mendapatkan informasi Detail Objek WiFi
Tujuan	Untuk mendapatkan informasi detail objek WiFi
Deskripsi	<i>Use case</i> ini digunakan untuk mendapatkan informasi lokasi yang dimaksud
Aktor	Pengguna
Pemicu	Pengguna memilih salah satu dari list rekomendasi yang ada
Kondisi Awal	Sistem menampilkan halaman rekomendasi
Skenario Utama (Basic Flow)	
Pengguna memilih salah satu <i>item</i> pada list tempat	

Kondisi Akhir	Sistem menampilkan detail informasi berupa nama, alamat, ulasan singkat, dan lokasi di peta.
----------------------	--

4.2.4 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang dibutuhkan oleh sistem. Ada beberapa parameter dan deskripsi kebutuhan yang akan digunakan dalam pengembangan, yaitu *usability* yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Spesifikasi Kebutuhan Non Fungsional

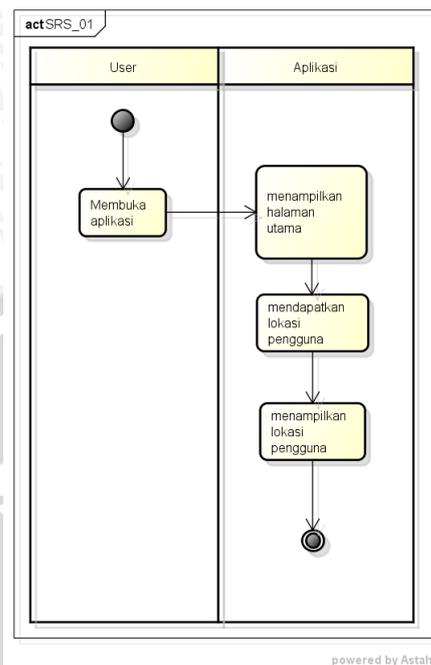
Parameter	Deskripsi Kebutuhan
<i>Usability</i>	Aplikasi harus dapat digunakan dengan mudah oleh pengguna. Aplikasi dapat membantu pengguna dalam memustuskan lokasi mana yang akan dituju serta memberikan informasi tentang lokasi yang ada.

4.3 Perancangan Perangkat Lunak

Setelah Dilakukan analisa kebutuhan, selanjutnya kebutuhan-kebutuhan tersebut dibentuk dalam suatu perancangan perangkat lunak yang terdiri dari *activity diagram*, *sequence diagram*, *class diagram*, perancangan basis data, dan perancangan antarmuka.

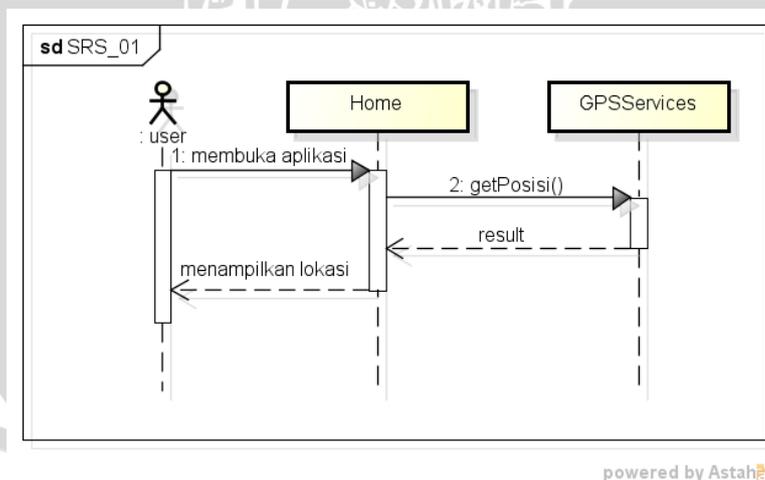
4.3.1 Mendapatkan Lokasi Pengguna

Sistem mendapatkan lokasi pengguna dan menampilkannya lewat *maps* saat aplikasi pertama dibuka. Dari Tabel skenario 4.2, maka didapatkan *activity diagram* yang ditunjukkan pada Gambar 4.3.



Gambar 4.3 Activity diagram mendapatkan lokasi pengguna

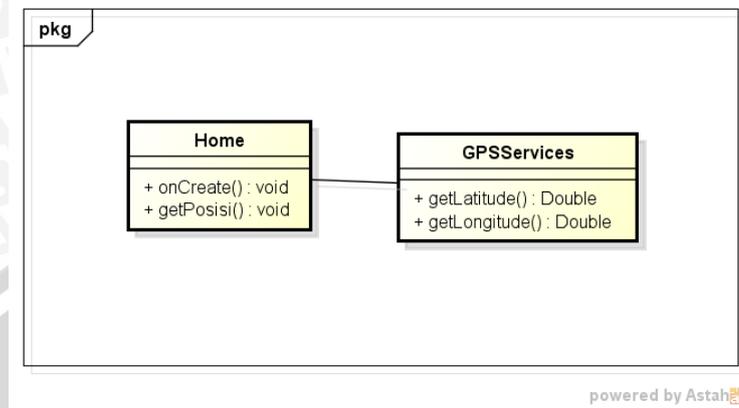
Pada *activity diagram* diatas pengguna cukup membuka aplikasi. Aplikasi kemudian akan melakukan proses untuk mendapatkan lokasi pengguna dan menampilkanya pada halaman utama. *Sequence diagram* untuk Gambar 4.3 diperlihatkan dalam Gambar 4.4.



Gambar 4.4 Sequence diagram mendapatkan lokasi pengguna

Saat pengguna membuka aplikasi, halaman *Home* meminta *latitude* dan *longitude* kepada kelas *GPSServices*. *GPSServices* kemudian memberikan *latitude*

dan *longitude* pengguna. Lokasi pengguna ditampilkan dalam *maps*. *Class diagram* pada Gambar 4.4 dijelaskan pada Gambar 4.5.



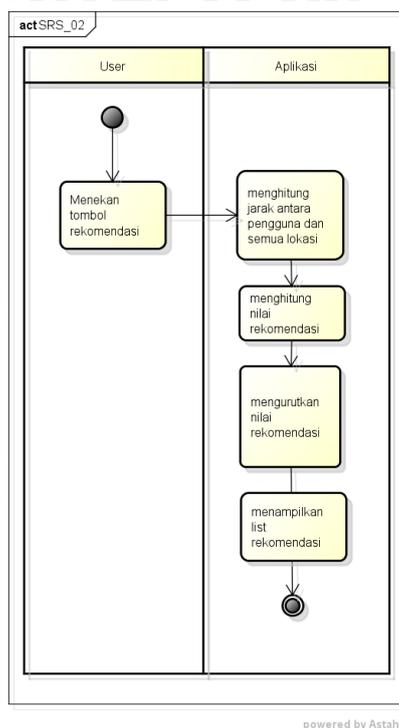
Gambar 4.5 *Class diagram* mendapatkan lokasi pengguna

Pada *class Home* terdapat metode *getPosisi()* untuk mengambil *latitude* dan *longitude* yang diproses di *class GPSServices*.

4.3.2 Mendapatkan List Rekomendasi Lokasi

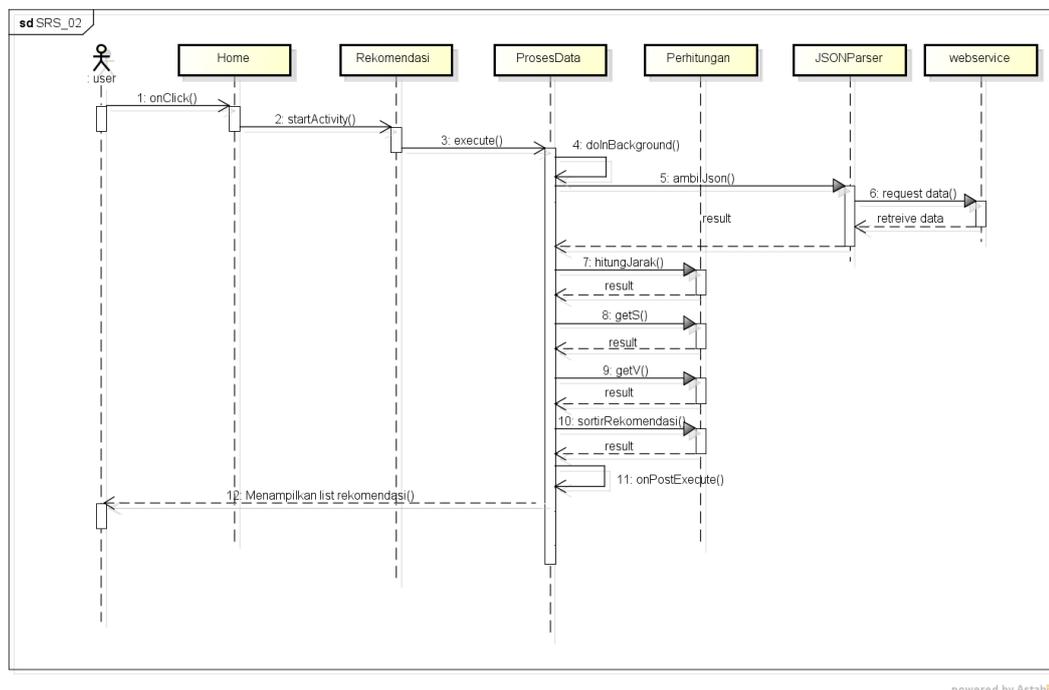
Pengguna bisa mendapatkan rekomendasi lokasi setelah sistem mendapatkan lokasi pengguna. *Activity diagram* mendapatkan list rekomendasi terdapat pada Gambar 4.6.





Gambar 4.6 Activity diagram mendapatkan list rekomendasi lokasi

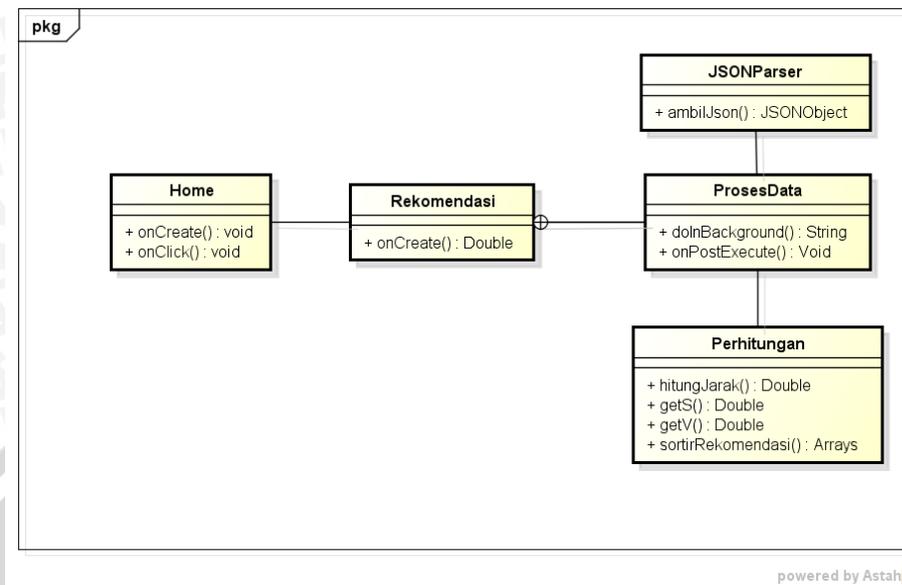
Pengguna cukup menekan tombol yang ada. Sistem kemudian melakukan perhitungan jarak dan rekomendasi serta menampilkannya pada pengguna. *Sequence diagram* untuk untuk Gambar 4.6 terdapat pada Gambar 4.7.



powered by Astah

Gambar 4.7 Sequence diagram mendapatkan list rekomendasi lokasi

Saat pengguna menekan tombol rekomendasi di halaman utama, sistem menjalankan aktifitas yang ada di kelas rekomendasi. Sistem mengakses *database* di *webservice* menggunakan kelas *JSONParser*. Pemrosesan data dilakukan di kelas *ProsesData*, sedangkan perhitungan untuk mendapatkan jarak dan nilai rekomendasi dilakukan di kelas *perhitungan*. Sehingga didapatkan *class diagram* seperti pada Gambar 4.8.

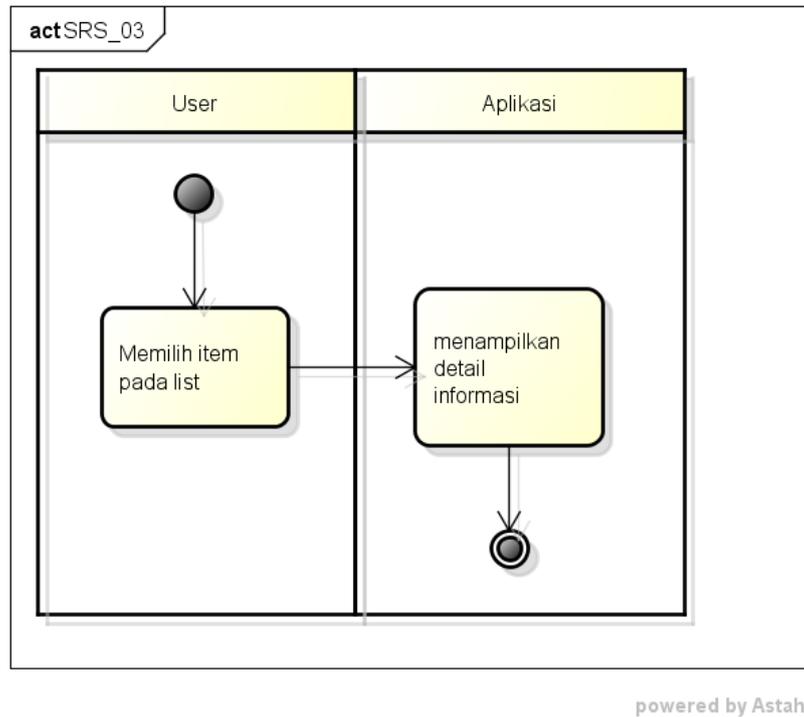


Gambar 4.8 Class diagram mendapatkan list rekomendasi lokasi

Kelas Rekomendasi melakukan eksekusi pada inner class *ProsesData*. Kelas *ProsesData* kemudian mengambil data dari *database* melalui *JSONParser* dan melakukan perhitungan melalui kelas *perhitungan*. Sistem kemudian menampilkan list rekomendasi kepada pengguna.

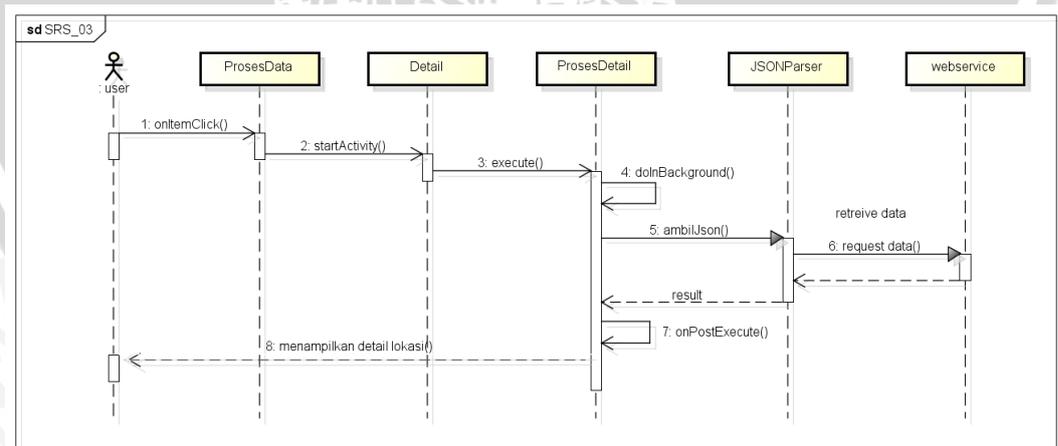
4.3.4 Mendapatkan Informasi Detail Objek WiFi

Setelah menampilkan list rekomendasi, pengguna bisa mendapatkan informasi tentang lokasi pada *item* yang dipilih. *Activity diagram* mendapatkan informasi lokasi terdapat pada Gambar 4.9.



Gambar 4.9 Mendapatkan informasi detail objek WiFi

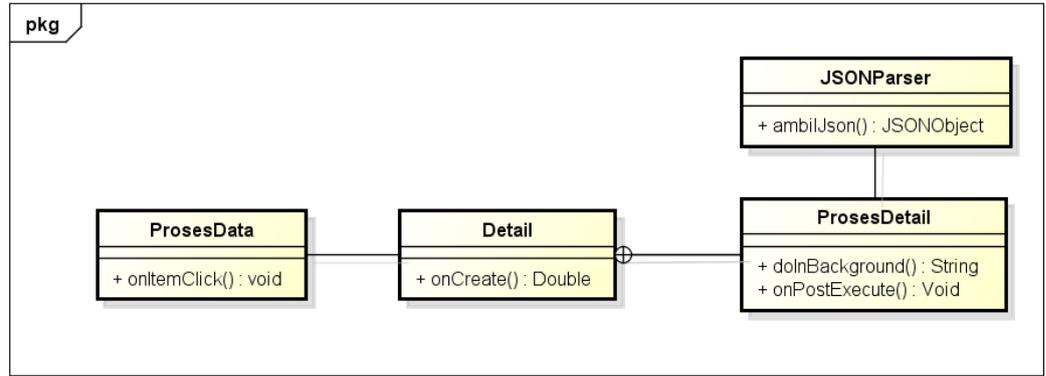
Pengguna cukup memilih salah satu *item* dari list rekomendasi. Sistem kemudian menampilkan informasi berupa nama, alamat, ulasan tentang lokasi, dan letak lokasi di *maps*. *Sequence diagram* mendapatkan informasi detail objek WiFi terdapat pada Gambar 4.10.



Gambar 4.10 *Sequence diagram* mendapatkan informasi detail objek WiFi

Saat pengguna menekan salah *item* pada list, kelas *ProsesDetail* memulai aktifitas kelas *Detail*. Kelas *ProsesDetail* mengambil data yang akan ditampilkan

dari *webservices* melalui *JSONParser*. Sistem kemudian menampilkan informasi lokasi pada pengguna. *Class diagram* mendapatkan informasi detail objek WiFi terdapat pada Gambar 4.11.



Gambar 4.11 *Class diagram* mendapatkan informasi detail objek WiFi

Saat pengguna memilih salah *item* rekomendasi, kelas *ProsesData* menjalankan aktifitas yang ada di kelas *Detail*. Kelas detail kemudian mengeksekusi inner class *ProsesDetail*. Sistem kemudian menampilkan informasi yang didapat pada kelas *ProsesDetail* pada pengguna.

4.3.5 Perancangan Basis Data

Basis data berfungsi untuk menyimpan data dan informasi yang diperlukan sistem. Sistem ini menggunakan sebuah Tabel untuk menyimpan data *marker* lokasi WiFi. Perancangan basis data pada aplikasi ini ditunjukkan pada Tabel 4.6.

Tabel 4.6 Struktur Tabel

No	Nama field	Type data	Keterangan	Contoh data	Kegunaan
1	Id	Numeric	Id lokasi	3, 15	Merupakan key untuk memanggil tiap <i>item</i> yang ada.
2	Nama	Text	Nama lokasi	Café Wilis, Roemah Moeria	Merupakan nama lokasi
3	Alamat	Text	Alamat lokasi	Jl. Wilis 200, Jl. Muria 20 Malang	Merupakan alamat lokasi
4	skor_kenya	Numeric	Skor	277, 365	Merupakan nilai

	manan		kenyamanan lokasi		kenyamanan lokasi
5	kecepatan_Internet	Numeric	Rata-rata kecepatan Internet lokasi dalam KBps	114, 100	Merupakan rata-rata kecepatan Internet lokasi
6	Biaya	Numeric	Biaya minimal yang dikeluarkan pengguna di lokasi	5000, 7500	Merupakan biaya minimal yang dikeluarkan pengguna di lokasi
7	<i>Longitude</i>	Numeric	Koordinat <i>longitude</i> jalan	110.325819, 112.318813	Merupakan <i>longitude</i> lokasi
8	<i>Latitude</i>	Numeric	Koordinat <i>latitude</i> jalan	-5.672311, -7.57335	Merupakan <i>latitude</i> lokasi
9	Informasi	Text	Ulasan lokasi	“Lokasi ini menawarkan menu cemilan dengan harga kisaran 1000 – 5000 rupiah. Di lokasi ini terdapat Musholla dan toilet yang bersih..”	Merupakan ulasan singkat tentang rentang harga dan fasilitas yang ada di lokasi

Data lokasi disimpan dalam *database* ini. *Longitude* dan *latitude* lokasi dihitung dengan *longitude* dan *latitude* lokasi pengguna sehingga menghasilkan jarak. Jarak, kecepatan_Internet, skor_kenyamanan, dan biaya diolah oleh sistem sehingga didapatkan skor rekomendasi. *Longitude* dan *latitude* juga digunakan untuk menampilkan lokasi WiFi yang dipilih pada *Maps*. Kolom alamat, nama, dan informasi ditampilkan di halaman informasi tempat.

4.3.6 Perancangan Antarmuka

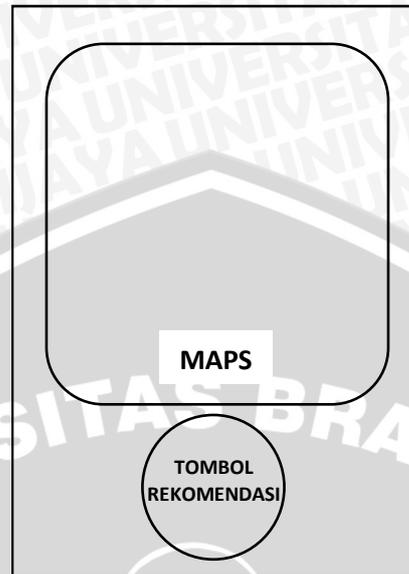
Perancangan antarmuka berisi rancangan tampilan aplikasi yang dibuat. Perancangan ini berfungsi agar aplikasi yang dibuat dapat dengan mudah digunakan oleh pengguna. Rancangan alur rekomendasi dan antar muka diperlihatkan pada Gambar 4.12.



Gambar 4.12 Rancangan alur proses dan antarmuka

4.3.6.1 Desain Antarmuka Halaman Utama Aplikasi

Halaman utama merupakan halaman yang muncul ketika aplikasi dibuka. Halaman utama mempunyai sebuah menu untuk mendapatkan rekomendasi lokasi dan sebuah tampilan peta untuk menunjukkan lokasi pengguna berada. Rancangan halaman utama aplikasi ditunjukkan pada Gambar 4.13.



Gambar 4. 13 Rancangan antarmuka halaman utama

Untuk mendapatkan rekomendasi lokasi, pengguna cukup menekan tombol rekomendasi. Keterangan Gambar 4.13 ditunjukkan pada Tabel 4.7.

Tabel 4. 7 Keterangan desain antarmuka halaman utama

Nama	Keterangan
Tombol Rekomendasi	Tombol ini merupakan tombol untuk mendapatkan rekomendasi lokasi. Saat pengguna menekan tombol ini, sistem akan melakukan perhitungan untuk mendapatkan jarak yang digunakan untuk menghitung nilai rekomendasi. Pengguna kemudian akan masuk ke halaman hasil rekomendasi yang berbentuk list rekomendasi lokasi.
<i>Maps</i>	Merupakan peta yang menunjukkan lokasi pengguna saat membuka aplikasi.

4.3.6.2 Desain Antarmuka Halaman Rekomendasi

Halaman hasil rekomendasi berisi list lokasi yang menjadi rekomendasi setelah pengguna menekan tombol untuk mendapatkan rekomendasi. Rancangan antarmuka halaman hasil rekomendasi ditunjukkan sebagai pada Gambar 4.14.



Gambar 4.14 Antar muka halaman rekomendasi

Untuk mendapatkan informasi lokasi, pengguna cukup menekan hasil rekomendasi yang ada. Halaman ini berisi beberapa list rekomendasi yang diurutkan dari skor rekomendasi tertinggi. Keterangan pada Gambar 4.14 akan ditunjukkan pada Tabel 4.8.

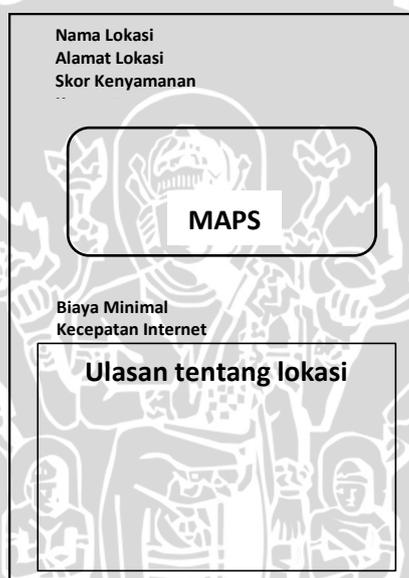
Tabel 4.8 Keterangan desain antarmuka halaman rekomendasi

Nama	Keterangan
Rekomendasi	Merupakan <i>item</i> lokasi yang menjadi rekomendasi sistem. Pengguna dapat memilih salah satu <i>item</i> dari list yang ketika dipilih akan membawa pengguna ke halaman informasi lokasi sesuai <i>item</i> lokasi yang dipilih.
Alamat	Merupakan alamat dari <i>item</i> lokasi yang menjadi rekomendasi.

Skor Rekomendasi	Merupakan nilai rekomendasi yang direpresentasikan dalam bentuk bintang.
Jarak	Merupakan jarak antara pengguna dan lokasi

4.3.6.3 Desain Antarmuka Halaman Detail

Halaman detail berisi informasi tentang lokasi yang dipilih pengguna di halaman hasil rekomendasi. Rancangan antarmuka halaman hasil rekomendasi digambarkan pada Gambar 4.15.



Gambar 4. 15 Antarmuka halaman detail

Penjelasan tentang desain antarmuka pada Gambar 4.15 dijelaskan pada Tabel 4.9.

Tabel 4. 9 Keterangan desain antarmuka halaman detail

Nama	Keterangan
Nama Lokasi	Merupakan nama dari <i>item</i> lokasi yang dipilih pada halaman hasil rekomendasi
Alamat Lokasi	Merupakan alamat dari <i>item</i> lokasi yang dipilih pada

	halaman hasil rekomendasi
Ulasan Lokasi	Merupakan ulasan dari <i>item</i> lokasi yang dipilih pada halaman hasil rekomendasi. Ulasan ini berisi artikel singkat, fasilitas yang ada, kecepatan Internet, dan biaya yang harus dikeluarkan.
Maps	Merupakan peta tempat lokasi tersebut berada
Skor Kenyamanan	Merupakan nilai kenyamanan yang direpresentasikan dalam bentuk bintang
Biaya Minimal	Merupakan biaya minimal pada menu atau minimum pembelian untuk mendapatkan akses WiFi
Kecepatan Internet	Merupakan kecepatan Internet lokasi

4.4 Proses Perhitungan

Ada 2 langkah perhitungan yang digunakan pada aplikasi ini. Yaitu perhitungan jarak pengguna ke lokasi dan perhitungan rekomendasi. Perhitungan jarak digunakan untuk mencari jarak pengguna ke lokasi. Hasil perhitungan ini kemudian diproses untuk menghitung rekomendasi.

4.4.1 Perhitungan Jarak

Perhitungan jarak dilakukan setelah pengguna menekan tombol rekomendasi. Perhitungan jarak memerlukan *longitude* dan *latitude* pengguna yang dikirim dari halaman utama serta *longitude* dan *latitude* lokasi yang didapatkan dari *database*. Perhitungan jarak antara 2 koordinat (*latitude* dan *longitude*) menggunakan rumus Haversine. Perhitungan ini dilakukan saat skenario pada Tabel 4.3 dijalankan.

4.4.1.1 Contoh Perhitungan Rumus Haversine

Sebagai contoh, dimisalkan *latitude* dan *longitude* pengguna adalah -7.9633666 dan 112.6128482. Sedangkan *latitude* dan *longitude* lokasi yang dituju

adalah -7.970177 dan 112.627524. Dengan menggunakan rumus Haversine maka langkah-langkah menghitung jarak adalah sebagai berikut:

$$\phi_1 = -7.9633666$$

$$\phi_2 = -7.970177$$

$$\begin{aligned}\Delta\phi &= -7.970177 - (-7.9633666) \\ &= 0.0068104\end{aligned}$$

$$\begin{aligned}\Delta\lambda &= 112.627524 - 112.6128482 \\ &= 0.0146758\end{aligned}$$

$$r = 6371.1$$

$$d = 2 * 6371.1 \arcsin \sqrt{\left(\sin^2 \left(\frac{0.0068204}{2} \right) + \cos(-7.9633666) \cos(-7.970177) \sin^2 \left(\frac{0.0146758}{2} \right) \right)}$$

$$d = 1.784778645$$

Melalui perhitungan diatas didapatkan jarak antara pengguna dan lokasi yang dituju adalah sekitar 1.79 kilometer atau 1784.8 meter.

4.4.2 Perhitungan Rekomendasi

Setelah semua jarak pengguna dan semua lokasi yang ada di *database* didapatkan, jarak kemudian diurutkan dari yang terdekat ke yang terjauh. Jarak terdekat kemudian diambil sebagai salah satu kriteria untuk melakukan perankingan guna menghasilkan rekomendasi tempat. Metode untuk menghitung rekomendasi menggunakan *weighted product* (WP). Perhitungan ini dilakukan saat skenario pada Tabel 4.3 dijalankan.

Perhitungan rekomendasi menggunakan *weighted product* memerlukan bobot dan kriteria guna menghitung vektor S yang nantinya menghasilkan nilai preferensi untuk perankingan. Kriteria yang digunakan adalah jarak dalam satuan meter, kecepatan Internet dalam satuan *kilobyte per second*, biaya yang dikeluarkan, dan skor kenyamanan. Kecepatan Internet, biaya yang dikeluarkan, dan skor kenyamanan didapatkan dari hasil survei lapangan yang dilakukan oleh

peneliti. Parameter untuk skor kenyamanan didapatkan dari hasil survei yang telah dilakukan pada awal penelitian.

Proses perhitungan bobot yang digunakan pada aplikasi ini pertama-tama berasal dari hasil pembagian jumlah sample yang setuju tiap kriteria yang dimaksud dengan jumlah total sample. Hasil pembagian tersebut kemudian dibagi dengan total hasil pembagian tiap kriteria dan dikali 100%. Hasil akhir dari perhitungan tersebut kemudian digunakan sebagai bobot kriteria.

Diantara keempat bobot kriteria tersebut, jarak dan biaya yang dikeluarkan bernilai negatif karena kedua kriteria tersebut merupakan kriteria yang jika nilainya semakin besar akan semakin merugikan pengguna.

4.4.2.1 Contoh Perhitungan Manual

Terdapat empat kriteria yaitu skor kenyamanan (C1), jarak (C2), biaya (C3), kecepatan internet (C4), dan bobot tiap kriteria:

$$\text{Skor kenyamanan (B1)} = 0.23$$

$$\text{Jarak (B2)} = -0.21$$

$$\text{Biaya (B3)} = -0.17$$

$$\text{Kecepatan Internet (B4)} = 0.39$$

Tabel nilai alternatif di setiap kriteria:

Tabel 4.10 Tabel nilai alternatif

Alternatif	Kriteria			
	C1	C2	C3	C4
Lokasi 1	333	1700	6000	52
Lokasi 2	359	1300	8000	127
Lokasi 3	209	2000	8000	57

Maka sesuai dengan implementasi rumus (2-1) didapatkan rumus vektor S:

$$S_n = (C1^{B1})(C2^{B2})(C3^{B3})(C4^{B4})$$

$$S1 = (333^{0.23})(1700^{-0.21})(6000^{-0.17})(52^{0.39}) = 0.848639988$$

$$S2 = (359^{0.23})(1300^{-0.21})(8000^{-0.17})(127^{0.39}) = 1.232258358$$

$$S3 = (209^{0.23})(2000^{-0.21})(8000^{-0.17})(57^{0.39}) = 0.727249184$$

Implementasi untuk menghitung preferensi perankingan sesuai dengan rumus (2-4):

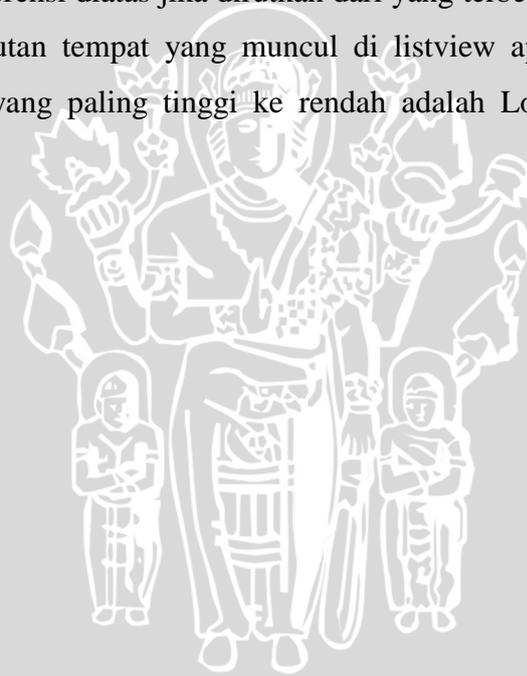
$$V_n = \frac{S_n}{\sum S}$$

$$V_1 = \frac{0.848639988}{0.848639988 + 1.232258358 + 0.727249184} = 0.30220634$$

$$V_2 = \frac{1.232258358}{0.848639988 + 1.232258358 + 0.727249184} = 0.438815392$$

$$V_3 = \frac{0.727249184}{0.848639988 + 1.232258358 + 0.727249184} = 0.258978268$$

Dari hasil preferensi diatas jika dirutkan dari yang terbesar adalah V2, V1, dan V3. Sehingga urutan tempat yang muncul di listview aplikasi sesuai dari tingkat rekomendasi yang paling tinggi ke rendah adalah Lokasi 2, Lokasi 1, kemudian Lokasi 3.



BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas hasil implementasi berdasarkan perancangan yang telah dibahas pada bab sebelumnya. Hasil implementasi tersebut kemudian diuji dan dianalisis yang nantinya akan dibahas pada sub bab pengujian dan analisis.

5.1 Lingkungan Implementasi

Lingkungan implementasi yang dijelaskan pada sub bab ini meliputi lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan untuk pengembangan maupun untuk implementasi aplikasi rekomendasi lokasi WiFi untuk mahasiswa berbasis Android.

5.1.1 Lingkungan Perangkat Keras

Lingkungan perangkat keras yang digunakan untuk pengembangan, instalasi, dan pengujian terdiri dari dua perangkat, yaitu laptop untuk pengembangan aplikasi dan perangkat bergerak (*smartphone*) untuk instalasi dan pengujian aplikasi. Perangkat keras yang digunakan untuk pengembangan aplikasi ditunjukkan pada Tabel 5.1.

Tabel 5. 1 Lingkungan Perangkat Keras Pengembangan Aplikasi

Nama Komponen	Spesifikasi
<i>System Model</i>	Dell Inspiron N4050
<i>Processor</i>	Intel® Core™ i5
<i>Memory (RAM)</i>	4096 MB
<i>Hardisk</i>	500 GB
<i>Display</i>	Intel® HD Graphic Family

Perangkat keras yang digunakan untuk instalasi dan pengujian aplikasi ditunjukkan pada Tabel 5.2.

Tabel 5. 2 Lingkungan Perangkat Keras Instalasi dan Pengujian Aplikasi

Nama Komponen	Spesifikasi
<i>System Model</i>	Sony Xperia M C2005 Samsung Galaxy Grand Duos
<i>Processor</i>	Qualcomm Dual-core Snapdragon S4 Plus MSM8227 1 GHz Dual-Core 1.2 Ghz Cortex-A9
<i>Memory</i>	4 GB, 1 GB RAM 8 GB, 1 GB RAM

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk pengembangan aplikasi ditunjukkan pada Tabel 5.3.

Tabel 5. 3 Lingkungan Perangkat lunak Pengembangan Aplikasi

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 7 Ultimate 64-bit
<i>IDE (Integrated Development Environment)</i>	Eclipse Kepler service release 2 dengan ADT (Android Development Tools) plug in
Perangkat Lunak Lain	CorelDraw X6 (64-bit) Xampp v3.2.1

Perangkat lunak yang digunakan untuk melakukan instalasi dan pengujian aplikasi ditunjukkan pada Tabel 5.4.

Tabel 5. 4 Lingkungan Perangkat Lunak Instalasi dan Pengujian Aplikasi

Nama Komponen	Spesifikasi
<i>Platform</i>	CyanogenMod 11 Android 4.4 (KitKat) Android 4.2 (Jelly Bean)

5.2 Batasan Implementasi

Dalam implementasi aplikasi rekomendasi lokasi WiFi *hotspot* berbasis Android terdapat beberapa batasan. Batasan tersebut antara lain :

1. Implementasi Google Map untuk menampilkan posisi pengguna maupun posisi lokasi yang dituju menggunakan Google *Maps* API versi 2.

2. Untuk menampilkan lokasi pengguna, *maps*, hasil rekomendasi, dan informasi tempat diperlukan koneksi Internet sehingga kecepatan akses untuk menampilkan *maps*, hasil rekomendasi, rute, dan informasi tempat bergantung pada kecepatan koneksi Internet pada masing-masing perangkat yang digunakan.

5.3 Implementasi Basis Data

Aplikasi rekomendasi lokasi WiFi menggunakan basis data untuk menyimpan data-data yang diperlukan. Struktur basis data yang ada dijelaskan sebelumnya pada Tabel 4.5. Implementasi basis data terhadap data riil ditunjukkan pada halaman lampiran.

5.4 Implementasi Class

Class-class yang telah dirancang pada bab perancangan diimplementasikan pada *file-file* dengan ekstensi *.java*. Tampilan dari *class-class* tersebut direalisasikan pada *file-file* dengan ekstensi *.xml*. Implementasi perancangan *class-class* yang berdasarkan Gambar 4.5, Gambar 4.8, dan Gambar 4.11 ditunjukkan pada Tabel 5.5.

Tabel 5. 5 Implementasi Perancangan *Class*

No	Package	Object	Nama Class	Nama File Program	Nama File Layout
1	com.rzlbs.dimanawifi	View, Controller	Home	Home.java	home.xml
2	com.rzlbs.dimanawifi	View, Controller	Detail	Detail.java	detail.xml
3	com.rzlbs.dimanawifi	View, Controller	Rekomendasi	Rekomendasi.java	rekomendasi.xml, single_rekomendasi.xml

4	com.rzlbs.dimanawifi	Model	JSONParser	JSONParser.java	-
5	com.rzlbs.dimanawifi	Controller	GPSServices	GPSServices.java	-
6	com.rzlbs.dimanawifi	Controller	Perhitungan	Perhitungan.java	-
7	com.rzlbs.dimanawifi	Controller, inner class	ProsesData	Rekomendasi.java	-
8	com.rzlbs.dimanawifi	Controller, inner class	ProsesDetail	Detail.java	-

5.5 Implementasi Kode Program

Pada implementasi Kode program akan dijelaskan hasil implementasi aplikasi sesuai dari perancangan yang telah dijelaskan pada bab sebelumnya. Perhitungan yang digunakan pada aplikasi rekomendasi lokasi Wi-Fi ini menggunakan 2 perhitungan yaitu Haversine untuk menghitung jarak dan *weighted product* untuk menghitung rekomendasi. Pada penulisan skripsi ini akan dijelaskan beberapa metode utama yang membangun aplikasi, sehingga tidak semua method dijelaskan pada bagian ini. Metode-metode tersebut antara lain method untuk perhitungan jarak menggunakan rumus Haversine dan perhitungan skor rekomendasi menggunakan *weighted product*.

5.5.1 Kode Program Mendapatkan Lokasi Pengguna

Bagian ini menjelaskan potongan Kode program dari implementasi dalam mendapatkan lokasi pengguna. Mendapatkan lokasi pengguna membutuhkan *longitude* dan *latitude* pengguna yang kemudian ditampilkan dalam bentuk

marker di *maps*. Implementasi Kode program mendapatkan lokasi pengguna ditunjukkan pada Kode 5.1.

```
public void getPosisi() {
    GPSServices posisi = new GPSServices(Home.this);
    lat_pengguna = posisi.getLatitude();
    lon_pengguna = posisi.getLongitude();
}
```

Kode 5.1 Implementasi Kode program mendapatkan lokasi pengguna

Metode `getPosisi` berisi variable `lat_pengguna` dan `lon_pengguna` untuk menyimpan *longitude* dan *latitude* dari pengguna. *Longitude* dan *latitude* pengguna yang didapatkan merupakan nilai kembalian dari metode `getLatitude` dan `getLongitude` yang ada di kelas `GPSServices`. Kemudian `lat_pengguna` dan `lon_pengguna` ditampilkan kepada pengguna dalam bentuk *marker* di dalam *maps*. Implementasi Kode program untuk menampilkan lokasi pengguna ditunjukkan pada Kode 5.2.

```
1 GoogleMap map = ((MapFragment)
2 getFragmentManager().findFragmentById(R.id.map)).getMap();
3 final LatLng LOKASI = new LatLng(lat_pengguna, lon_pengguna);
4 Marker lokasi = map.addMarker(newMarkerOptions().position(LOKASI)
5 .title("Lokasi Anda"));
6 map.moveCamera(CameraUpdateFactory.newLatLngZoom(LOKASI, 15));
7 map.animateCamera(CameraUpdateFactory.zoomTo(15), 2000, null);
```

Kode 5.2 Implementasi Kode program menampilkan lokasi pengguna

Variabel `map` merupakan deklarasi untuk menampilkan *Google Maps* Api ke fragment yang tersedia pada layout. Kemudian variable `LOKASI` mendeklarasikan *longitude* dan *latitude* pengguna yang sebelumnya telah didapatkan. Variabel `lokasi` pada line 4 berguna untuk menampilkan *marker* di *maps*. Kode program `LOKASI` pada line 6 dan 7 merupakan Kode program untuk mengarahkan tampilan langsung menuju variable `LOKASI` yang telah dideklarasikan sebelumnya dan melakukan zoom sesuai nilai yang ada.

5.5.2 Kode Program Mendapatkan List Rekomendasi Lokasi

Bagian ini menjelaskan potongan Kode program untuk perhitungan yang diimplementasikan pada aplikasi rekomendasi lokasi Wi-Fi. Perhitungan yang digunakan mencakup 2 metode yaitu metode Haversine untuk perhitungan jarak

antara pengguna dan semua lokasi yang ada di *database* dan metode *weighted product* untuk menghitung skor rekomendasi.

Nilai *longitude* dan *latitude* pengguna yang sebelumnya telah didapatkan kemudian dikirimkan dari kelas Home ke kelas Rekomendasi. Pengiriman dan penerimaan nilai tersebut ditunjukkan pada Kode 5.3 dan 5.4.

```
public void onClick(View v) {
    Intent intent = new Intent(Home.this,
    Rekomendasi.class);
    Bundle bundle = new Bundle();
    bundle.putDouble("lat_pengguna", lat_pengguna);
    bundle.putDouble("lon_pengguna", lon_pengguna);
    intent.putExtras(bundle);
    startActivity(intent);
}
```

Kode 5.3 Pengiriman nilai *longitude* dan *latitude*

```
Bundle bundle = getIntent().getExtras();
lat_pengguna = bundle.getDouble("lat_pengguna");
lon_pengguna = bundle.getDouble("lon_pengguna");
```

Kode 5.4 Penerimaan nilai *longitude* dan *latitude*

Aplikasi kemudian akan mengeksekusi inner class mengambil *longitude* dan *latitude* yang ada pada *database*. Implementasi Kode program untuk pengambilan *longitude* dan *latitude* yang ada di *database* ditunjukkan pada Kode 5.5.

```
JSONParser jParser = new JSONParser();
JSONObject json = jParser.AmbilJson(link_url);
try {
    dimana_wifi = json.getJSONArray("list_lokasi");
    r = dimana_wifi.length();
    araiWP = new double[r][3];
    hasilWP = new String[r][1];

    for (int i = 0; i < r; i++) {
        JSONObject dw = dimana_wifi.getJSONObject(i);
        String id = dw.getString(DW_ID);
        String longi = dw.getString(DW_LONGI);
        String lati = dw.getString(DW_LATI);
        String kenyamanan = dw.getString("skor_kenyamanan");
        String biaya = dw.getString("biaya");
        String kecepatan = dw.getString("kecepatan_Internet");

        double lon_lokasi = Double.parseDouble(longi);
        double lat_lokasi = Double.parseDouble(lati);
        double ken_lokasi = Double.parseDouble(kenyamanan);
        double bia_lokasi = Double.parseDouble(biaya);
        double kec_lokasi = Double.parseDouble(kecepatan);
    }
}
```

```

        double jarak = perhitungan.hitungJarak(lon_pengguna,
lat_pengguna,
lon_lokasi, lat_lokasi);
        araiWP[i][0] = Double.parseDouble(id);
        araiWP[i][1] = jarak;
        araiWP[i][2] = perhitungan.getS(ken_lokasi, jarak,
            bia_lokasi, kec_lokasi);
    }
} catch (JSONException e) {
    e.printStackTrace();
}

```

Kode 5.5 Mengambil *longitude* dan *latitude* dari database

Aplikasi akan mengambil id, *longitude*, *latitude*, skor kenyamanan, biaya, dan kecepatan Internet yang ada di *database*. Pengambilan data tersebut melalui kelas *JSONParser* oleh variable *link_url* yang telah dideklarasikan sebelumnya. *Longitude*, *latitude*, skor kenyamanan, biaya, dan kecepatan Internet yang bertipe data *String* kemudian dikonversi menjadi *double*. Program kemudian akan menghitung jarak melalui metode *hitungJarak*. Hasil dari perhitungan jarak tersebut disimpan dalam variable *jarak*. Aplikasi kemudian akan menghitung vektor *S* melalui metode *getS*.

Perhitungan jarak yang ada menggunakan rumus Haversine. Implementasi Kode program untuk rumus Haversine ditunjukkan pada Kode 5.6.

```

1 public double hitungJarak(double lonPos, double latPos, double
2 lonLoc, double latLoc) {
3     double deltalon, deltalat, lat1, lat2, jarak = 0;
4
5     deltalon = deg2rad(lonLoc - lonPos);
6     deltalat = deg2rad(latLoc - latPos);
7     lat1 = deg2rad(latPos);
8     lat2 = deg2rad(latLoc);
9     jarak = Math.sin(deltalat / 2) * Math.sin(deltalat / 2) +
10 Math.sin(deltalon / 2) * Math.sin(deltalon / 2) * Math.cos(lat1) *
11 Math.cos(lat2);
12     jarak = 2 * Math.asin(Math.sqrt(jarak));
13     jarak = jarak * 6371.1 * 1000;
14     BigDecimal bd = new BigDecimal(jarak);
15     bd = bd.setScale(2, BigDecimal.ROUND_UP);
16     jarak = bd.doubleValue();
17
18     return jarak;
19 }

```

Kode 5.6 Implementasi Metode Haversine

Variabel `deltaLon` dan `deltaLat` merupakan selisih *longitude* dan *latitude* antara pengguna dan lokasi yang dituju. Variabel `jarak` pada line 9 merupakan implementasi awal metode Haversine. Hasil perhitungan tersebut kemudian dihitung akarnya, Hasil perhitungan akar tersebut kemudian dicari nilai arcsin dan dikali 2. Hasil dari perhitungan tersebut kemudian dikali 6371.1 untuk mengetahui jarak dalam kilometer. Perkalian dengan angka 1000 berguna untuk mengkonversi satuan kilometer ke meter. Hasil dari perhitungan tersebut kemudian dibulatkan hingga 2 angka di belakang koma dan menghasilkan jarak pengguna ke seluruh lokasi yang terdaftar di *database* dalam satuan meter.

Metode `deg2rad` merupakan metode untuk mengkonversi nilai derajat ke radian.. Metode `deg2rad` ditunjukkan pada Kode 5.7.

```
private double deg2rad(double deg) {
    return (deg * Math.PI / 180.0);
}
```

Kode 5.7 Metode Konversi Derajat ke Radian

Setelah jarak didapatkan, program kemudian melakukan proses perhitungan rekomendasi. Selain jarak, perhitungan rekomendasi yang ada membutuhkan kecepatan internet, skor kenyamanan, dan biaya minimal yang dikeluarkan. Ketiga nilai kriteria tersebut diambil dari *database*.

Nilai yang ada kemudian dihitung di metode `getS` untuk vektor `S`. Implementasi Kode program untuk mendapatkan vektor `S` ditunjukkan pada Kode 5.8.

```
public double getS(double kenyamanan, double jaraks,
double biaya, double kecepatan) {
    double pangkatNyaman = 0.316;
    double pangkatJarak = -0.176;
    double pangkatBiaya = -0.199;
    double pangkatCepat = 0.309;
    double s;

    s = Math.pow(kenyamanan, pangkatNyaman)
        * Math.pow(jaraks, pangkatJarak)
        * Math.pow(biaya, pangkatBiaya)
        * Math.pow(kecepatan, pangkatCepat);

    return s;
}
```

Kode 5.8 Kode program menghitung vektor S

Vektor S membutuhkan bobot kriteria yang telah didapatkan melalui hasil dari penelitian sebelumnya. Nilai dari bobot tersebut disimpan di variable pangkatNyaman, pangkatJarak, pangkatBiaya, dan pangkatCepat. Nilai dari kriteria yang ada kemudian dipangkatkan dengan bobot kriteria masing-masing. Hasil dari perhitungan tersebut menghasilkan vektor S yang nilainya disimpan di variable s.

Hasil dari perhitungan vektor S tersebut kemudian dijumlahkan dan menjadi pembagi untuk mendapatkan nilai perankingan. Implementasi Kode program untuk mendapatkan nilai pembagi perankingan ditunjukkan pada Kode 5.9.

```
for (int p = 0; p < r; p++) {
    pembagi = pembagi + araiWP[p][2];
}
for (int l = 0; l < r; l++) {
    araiWP[l][2] = perhitungan.getV(araiWP[l][2], pembagi);
}
araiHWP = perhitungan.sortirRekomendasi(araiWP, r);
```

Kode 5.9 Mendapatkan nilai pembagi perankingan

Nilai pada perhitungan jarak yang telah disimpan pada variable araiWP kemudian dijumlahkan untuk mendapatkan pembagi. Tiap nilai dari araiWP kemudian dibagi dengan pembagi untuk mendapatkan skor perankingan melalui metode getV. Nilai akhir tersebut kemudian disortir melalui metode sortirRekomendasi untuk mengurutkan nilai dari yang terbesar. Nilai perankingan tersebut Perhitungan untuk mendapatkan skor perankingan ditunjukkan pada Kode 5.10.

```
public double getV(double finalS, double totalS) {
    double totalOfV = finalS / totalS;
    return totalOfV;
}
```

Kode 5.10 Kode program skor perankingan

Variabel totalOfV merupakan hasil dari perhitungan perankingan. Tiap nilai tersebut kemudian disimpan di variable array araiJarak. Variabel tersebut kemudian diurutkan melalui sortirRekomendasi sehingga menghasilkan nilai rekomendasi dengan urutan dari besar ke kecil.

5.5.3 Kode Program Mendapatkan Informasi Detail Objek WiFi

Bagian ini menjelaskan potongan Kode program untuk mendapatkan informasi detail objek WiFi. Hasil dari perhitungan rekomendasi sebelumnya ditampilkan dalam bentuk list. Potongan Kode program untuk menampilkan list rekomendasi ditunjukkan pada Kode 5.11.

```
ListAdapter adapter = new SimpleAdapter(Rekomendasi.this,
    daftar_tempat, R.layout.single_rekomendasi,
    new String[] {DW_NAMA, DW_ALAMAT, DW_ID, skornya }, new int[] {
        id.nama, R.id.alamat, R.id.Kode_id, R.id.skorwp });
setListAdapter(adapter);
```

Kode 5. 11 List Rekomendasi

Nilai dari tiap hasil dari perhitungan rekomendasi sebelumnya disimpan dalam sebuah variabel hashmap. Variabel tersebut kemudian di panggil di dalam SimpleAdapter dengan mendeklarasikan layout, key, serta id layout. Variabel key tersebut adalah DW_NAMA yang menyimpan nama, DW_ALAMAT yang menyimpan alamat, DW_ID yang menyimpan id, serta variable skornya yang menyimpan skor rekomendasi. Variabel dengan nilai-nilai tersebut kemudian disimpan di dalam variable adapter dan ditampilkan melalui setListAdapter.

Nilai id yang ada nantinya akan dikirm untuk mendapatkann informasi lokasi dari *item* yang dipilih. Pengiriman nilai id ditunjukkan pada Kode 5.12. Penerimaan nilai untuk menampilkan informasi lokasi ditunjukkan pada Kode 5.13.

```
public void onItemClick(AdapterView<?> parent, View view,
    int position, long id) {
    String Kode_id = ((TextView) view
        .findViewById(R.id.Kode_id)).getText().toString();
    Intent intent = new Intent(Rekomendasi.this,Detail.class);
    intent.putExtra(DW_ID, Kode_id);
```

Kode 5. 12 Pengiriman nilai Id

```
Intent in = getIntent();
String Kode = in.getStringExtra(DW_ID);
link_url = "http://dimanawifi.wc.lt/detail_lokasi.php?Kode=" +
Kode;
```

Kode 5. 13 Penerimaan nilai untuk menampilkan informasi

Item yang dipilih diproses oleh metode onItemClick. Variabel Kode_id kemudian akan menyimpan id dari *item* yang dipilih dan dikirimkan dengan key

DW_ID. Nilai tersebut kemudian dikirim ke halaman detail dan disimpan dalam variable Kode. Nilai dalam variable tersebut digunakan untuk mendeklarasikan variable link_url yang berupa url tempat JSON yang dibutuhkan berada. Data dalam JSON tersebut kemudian ditampilkan kepada pengguna.

5.5.4 Kode Program JSON

Pengambilan data pada *database* dilakukan melalui kode program yang diletakan pada server. Kode program tersebut menghasilkan data berupa JSON yang kemudian diolah oleh sistem sehingga menghasilkan rekomendasi lokasi. Kode 5.14 merupakan kode program yang diproses untuk mendapatkan nilai rekomendasi yang disimpan dalam file bernama tempatWifi dengan format php.

```
<?php
error_reporting(0);
require("koneksi.php");
$query = "select * from main_table";

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = "Database Error!";
    die(json_encode($response));
}
$rows = $stmt->fetchAll();

if ($rows) {
    $response["success"] = 1;
    $response["message"] = "Post Available!";
    $response["list_lokasi"] = array();

    foreach ($rows as $row){
        $post = array();
        $post["id"] = $row["id"];
        $post["longitude"] = $row["longitude"];
        $post["latitude"] = $row["latitude"];
        $post["skor_kenyamanan"] = $row["skor_kenyamanan"];
        $post["kecepatan_internet"] = $row["kecepatan_internet"];
        $post["biaya"] = $row["biaya"];
        array_push($response["list_lokasi"], $post);
    }
    echo json_encode($response);
} else {
    $response["success"] = 0;
    $response["message"] = "No Post Available!";
    die(json_encode($response));
}
```

```
?>
```

Kode 5.14 Mengambil Data untuk Perhitungan Rekomendasi

Data yang dibutuhkan disimpan dalam variabel `id`, `longitude`, `latitude`, `skor_kenyamanan`, `kecepatan_internet`, dan `biaya`. Kemudian variable-variabel tersebut disimpan dalam array bernama `list_lokasi`. Data-data yang ada kemudian digunakan untuk menghitung jarak serta nilai rekomendasi yang nanti menghasilkan list rekomendasi.

Kode 5.15 merupakan kode program yang diproses untuk menampilkan detail objek lokasi WiFi.

```
<?php
error_reporting(0);
$kd = $_GET['kode'];
require("koneksi.php");
$query = "select * from main_table where id=".$kd;

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = "Database Error!";
    die(json_encode($response));
}
$rows = $stmt->fetchAll();

if ($rows) {
    $response["success"] = 1;
    $response["message"] = "Post Available!";
    $response["list_lokasi"] = array();

    foreach ($rows as $row) {
        $post = array();
        $post["id"] = $row["id"];
        $post["nama"] = $row["nama"];
        $post["alamat"] = $row["alamat"];
    }
}
```

```
$post["informasi"] = $row["informasi"];
$post["longitude"] = $row["longitude"];
$post["latitude"] = $row["latitude"];
$post["skor_kenyamanan"] = $row["skor_kenyamanan"];
$post["kecepatan_internet"] = $row["kecepatan_internet"];
$post["biaya"] = $row["biaya"];

array_push($response["list_lokasi"], $post);
}
echo json_encode($response);}
else {
    $response["success"] = 0;
    $response["message"] = "No Post Available!";
    die(json_encode($response));
}

?>
```

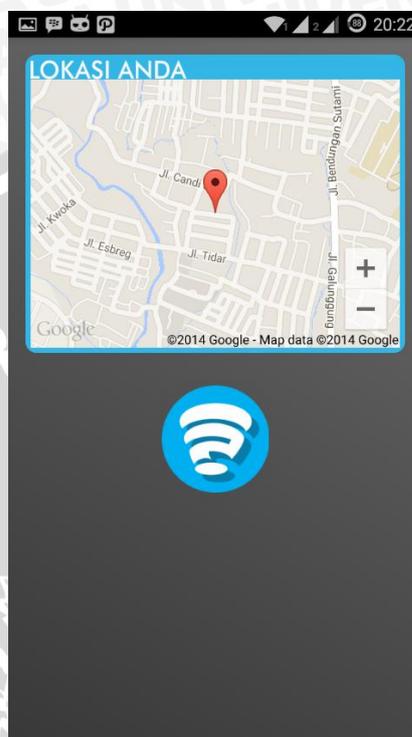
Data yang dibutuhkan disimpan dalam variabel `id`, `nama`, `alamat`, `informasi`, `longitude`, `latitude`, `skor_kenyamanan`, `kecepatan_internet`, dan `biaya`. Data-data tersebut merupakan data yang ada dalam *database*. Kemudian variable-variabel tersebut disimpan dalam array bernama `list_lokasi`. Data-data yang ada kemudian digunakan untuk menampilkan informasi detail pada lokasi WiFi yang dipilih pengguna pada halaman rekomendasi.

5.6 Implementasi Antarmuka

Bagian ini menjelaskan implementasi antarmuka berdasarkan perancangan yang telah dilakukan sebelumnya.

5.6.1 Implementasi Antarmuka Halaman Utama

Halaman utama merupakan halaman yang muncul pertama kali setelah pengguna membuka aplikasi. Implementasi antarmuka halaman utama berdasarkan pada Gambar 4.13 diperlihatkan pada Gambar 5.1.

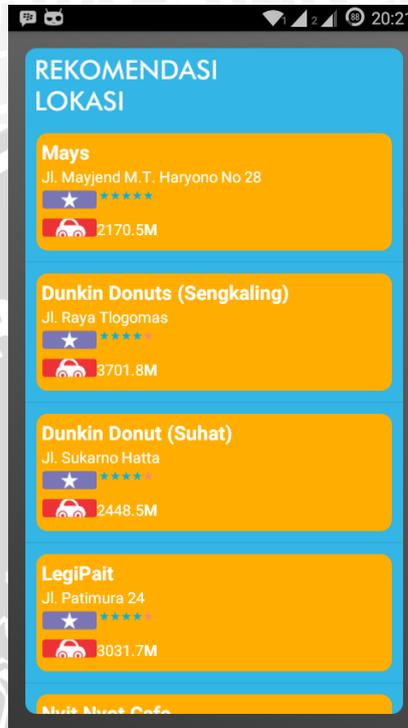


Gambar 5.1 Implementasi Antarmuka Halaman Utama

Halaman utama berisi 2 elemen, yaitu peta lokasi pengguna berada dan tombol untuk mendapatkan rekomendasi. Ketika tombol untuk mendapatkan rekomendasi ditekan, pengguna akan diarah menuju halaman rekomendasi.

5.6.2 Implementasi Antarmuka Halaman Rekomendasi

Halaman rekomendasi merupakan halaman yang berisi list rekomendasi lokasi. Implementasi antarmuka halaman rekomendasi berdasarkan Gambar 4.14 diperlihatkan pada Gambar 5.2.



Gambar 5.2 Implementasi Antarmuka halaman rekomendasi

Tiap *item* dari list yang direkomendasikan berisi nama lokasi, alamat lokasi, dan skor dari hasil perankingan yang direpresentasikan dalam jumlah bintang. Semakin banyak jumlah bintang yang ada maka skor perankingannya lebih besar.

5.6.3 Implementasi Antarmuka Halaman Detail

Halaman detail merupakan halaman yang berisi detail ulasan serta lokasi dari salah satu *item* yang dipilih pada halaman rekomendasi. Implementasi antarmuka halaman detail berdasarkan Gambar 4.15 diperlihatkan pada Gambar 5.3.



Gambar 5. 3 Antarmuka halaman detail

Halaman detail berisi tentang nama, alamat, ulasan dan peta lokasi serta rute yang bisa ditempuh.

5.7 Pengujian

Bagian ini berisi tentang pengujian yang akan dilakukan pada aplikasi yang telah dibangun. Pengujian dilakukan dengan metode *BlackBox Testing*. Pengujian yang dilakukan antara lain pengujian validasi terhadap implementasi dari perancangan yang telah dilakukan dan pengujian akurasi sistem terhadap algoritma yang digunakan dengan prediksi dari perhitungan manual. Pengujian *usability* juga dilakukan untuk mengetahui nilai kegunaan aplikasi pada pengguna.

5.7.1 Pengujian Validasi Fitur

Pengujian validasi dilakukan untuk memastikan bahwa aplikasi yang dibangun telah memenuhi kebutuhan fungsional sesuai bab perancangan. Pengujian validasi merupakan pengujian yang tidak memperhatikan alur algoritma program atau bisa disebut dengan metode *BlackboxTesting*. Pengujian ini lebih ditekankan pada kesesuaian antara sistem yang telah dibangun dengan daftar kebutuhan fungsional. Berikut merupakan kasus uji mendapatkan lokasi pengguna yang ditunjukkan pada Tabel 5.6.

Tabel 5. 6 Kasus Uji Mendapatkan Lokasi Pengguna

Nomor Kasus Uji	UVL_01
Nama Kasus Uji	Kasus Uji Mendapatkan Lokasi Pengguna
Objek Uji	Kebutuhan Fungsional SRS_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat mendeteksi lokasi pengguna dan menampilkannya.
Data	Pengguna berada di sekitar Jl. Terusan Tidar Sakti
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi di sekitar Jl. Terusan Tidar Sakti. 2. Sistem menampilkan lokasi pengguna.
Hasil yang Diharapkan	Pengguna mendapatkan lokasi berada di sekitar Jl. Terusan Tidar Sakti

Berikut merupakan kasus uji mendapatkan rekomendasi lokasi lokasi yang ditunjukkan pada Tabel 5.7.

Tabel 5. 7 Kasus Uji Mendapatkan List Rekomendasi Lokasi

Nomor Kasus Uji	UVL_02
Nama Kasus Uji	Kasus Uji Mendapatkan Rekomendasi Lokasi
Objek Uji	Kebutuhan Fungsional SRS_02
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat menampilkan list rekomendasi pada pengguna.
Data	Pengguna berada di sekitar Jl. Terusan Tidar Sakti
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol yang ada di halaman

	<p>utama.</p> <ol style="list-style-type: none"> 2. Sistem melakukan perhitungan rekomendasi berdasarkan jarak, biaya, kecepatan Internet, dan nilai kenyamanan. 3. Sistem menampilkan list rekomendasi dengan rekomendasi teratas <i>Mays</i>.
Hasil yang Diharapkan	Pengguna mendapatkan list rekomendasi lokasi dengan rekomendasi teratas <i>Mays</i> .

Berikut merupakan kasus uji mendapatkan informasi detail objek WiFi yang ditunjukkan pada Tabel 5.8.

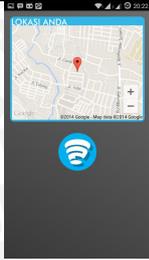
Tabel 5. 8 Kasus Uji Mendapatkan Informasi Detail Objek WiFi

Nomor Kasus Uji	UVL_03
Nama Kasus Uji	Kasus Uji Mendapatkan Informasi Detail Objek Wifi
Objek Uji	Kebutuhan Fungsional SRS_03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat menampilkan informasi lokasi pada pengguna.
Data	Pengguna memilih <i>item</i> list <i>Mays</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih <i>item</i> list <i>Mays</i> 2. Sistem menampilkan informasi tentang <i>Mays</i> berupa nama, alamat, peta lokasi, dan ulasan singkat
Hasil yang Diharapkan	Pengguna mendapatkan informasi tentang <i>Mays</i> berupa nama, alamat, peta lokasi, dan ulasan singkat

Tabel 5.9 merupakan hasil pengujian validasi berdasarkan kasus uji yang telah dibuat sebelumnya.

Tabel 5. 9 Hasil Pengujian Validasi

No.	Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validasi	Screenshot
-----	-----------------	-----------------------	-----------------------	-----------------	------------

1.	UVL_01	Pengguna mendapatkan lokasi tempat berdiri.	Pengguna mendapatkan lokasi tempat berdiri.	VALID	
2.	UVL_02	Pengguna mendapatkan list rekomendasi lokasi.	Pengguna mendapatkan list rekomendasi lokasi.	VALID	
3.	UVL_03	Pengguna mendapatkan informasi lokasi.	Pengguna mendapatkan informasi detail objek WiFi.	VALID	

5.7.2 Pengujian Akurasi Sistem

Pengujian ini berfungsi untuk mengetahui apakah algoritma untuk perhitungan jarak dan nilai rekomendasi yang diimplementasikan sudah benar. Data yang diuji berjumlah 2 data. Prosedur pengujian yang dilakukan adalah memasukan *latitude* dan *longitude* pengguna yang nantinya akan menghasilkan rekomendasi lokasi. Hasil rekomendasi tersebut kemudian dicocokkan dengan kesesuaiannya dengan prediksi dari perhitungan manual. Hasil pengujian akurasi sistem terhadap hasil rekomendasi ditunjukkan pada Tabel 5.10.

Tabel 5. 10 Hasil Pengujian Akurasi Sistem Terhadap Perhitungan Manual

No	Data Input		Angka Hasil	Urutan Perhitungan	Urutan Perhitungan Sistem	Status
	Latitude	Longitude	Perhitungan Manual	Manual		
1	-7.923441	112.576751	0.142839003	Dunkin Donuts (Sengkaling)	Dunkin Donuts (Sengkaling)	Akurat
			0.133489306	Mays	Mays	
			0.115244451	Dunkin Donuts (Suhat)	Dunkin Donuts (Suhat)	
			0.105420431	LegiPait	LegiPait	
			0.103893956	Republika	Republika	
			0.096519562	McDonalds	McDonalds	
			0.081704751	Indomaret Point	Indomaret Point	
			0.080603665	Nyit Nyot Café	Nyit Nyot Café	
			0.070146507	8oz	8oz	
			0.070138369	Kedai Kopi NKRI	Kedai Kopi NKRI	
2	-7.9643893	112.6088008	0.131306593	Mays	Mays	Akurat
			0.115106116	Dunkin Donuts	Dunkin Donuts (Sengkaling)	

		(Sengkaling)	
	0.11367719	Dunkin Donuts (Suhat)	Dunkin Donuts (Suhat)
	0.110103942	LegiPait	LegiPait
	0.104957339	Nyit Nyot Café	Nyit Nyot Café
	0.098384486	Republika	Republika
	0.097281516	McDonalds	McDonalds
	0.081997069	Indomaret Point	Indomaret Point
	0.078110443	Kedai Kopi NKRI	Kedai Kopi NKRI
	0.069075305	8oz	8oz

5.7.3 Pengujian Usability

Usability testing merupakan sebuah pengujian yang mengkaji seberapa mudah aplikasi digunakan. Pengujian ini menggunakan kuesioner yang dibagikan kepada 30 penguji secara acak. Pertanyaan yang diajukan dalam kuesioner ini mencakup daya guna aplikasi, kemudahan penggunaan, kemudahan dalam mempelajari aplikasi, dan kepuasan pengguna terhadap aplikasi. Pertanyaan-pertanyaan tersebut akan disertakan sebagai lampiran. Hasil dari pengujian *usability* ditunjukkan pada Tabel 5.11.

Tabel 5. 11 Hasil Pengujian *Usability*

No	Pernyataan	STS	TS	N	S	SS	Total
Daya Guna Aplikasi							
1	Aplikasi ini cukup membantu saya dalam memutuskan lokasi mana yang akan menjadi tujuan saya	0	0	0	10	20	30
2	Saya merasa aplikasi ini dapat membantu saya untuk mengetahui informasi lokasi yang akan saya tuju.	0	0	0	16	14	30
3	Aplikasi ini bermanfaat untuk saya.	0	0	2	15	13	30
Kemudahan Penggunaan							
4	Saya tidak perlu mengeluarkan banyak tenaga (menekan tombol, menggeser layar) dalam menggunakan aplikasi ini	0	0	2	11	17	30
5	Navigasi dalam aplikasi ini memudahkan saya dalam penggunaannya	0	0	3	12	15	30
6	Saya merasa aplikasi ini sangat mudah untuk digunakan.	0	0	1	13	16	30
Kemudahan Dalam Mempelajari Aplikasi							
7	Saya dengan cepat dan mudah	0	1	2	11	16	30

	mempelajari aplikasi ini tanpa membaca panduan atau bertanya terlebih dahulu						
8	Saya dengan cepat mengingat bagaimana cara menggunakan aplikasi ini	0	0	0	12	18	30
Kepuasan							
9	Saya menyukai tampilan antarmuka pada aplikasi ini.	0	1	5	14	10	30
10	Aplikasi ini berjalan sesuai dengan keinginan saya	0	0	6	11	13	30
11	Saya merasa puas dengan aplikasi ini	0	0	1	16	13	30

Keterangan:

STS : Sangat Tidak Setuju

S : Setuju

TS : Tidak Setuju

SS : Sangat Setuju

N : Netral

5.8 Hasil dan Analisis Pengujian

Setelah melalui tahap pengujian selanjutnya dilakukan analisis terhadap hasil dari pengujian yang dilakukan. Analisis dilakukan memperoleh kesimpulan dari hasil setiap pengujian yang dilakukan.

5.8.1 Analisis Hasil Pengujian Validasi Fitur

Analisis dari hasil pengujian validasi dilakukan dengan membandingkan kesesuaian hasil kinerja sistem yang telah dibangun dengan daftar kebutuhan yang telah didefinisikan pada tahap perancangan. Berdasarkan hasil pengujian yang telah dilakukan dapat diketahui bahwa hasil implementasi sistem telah sesuai dengan fitur-fitur pada daftar kebutuhan yang telah didefinisikan pada tahap perancangan.

5.8.2 Analisis Hasil Pengujian Akurasi Sistem

Analisis dari hasil pengujian akurasi sistem dilakukan dengan membandingkan kesesuaian perhitungan manual dengan hasil yang ditampilkan

oleh sistem. Berdasarkan hasil pengujian yang telah dilakukan dapat diketahui bahwa hasil perhitungan oleh sistem sesuai dengan hasil perhitungan manual.

5.8.3 Analisis Hasil Pengujian *Usability*

Proses analisis terhadap hasil pengujian *usability* dilakukan dengan menggunakan skala Likert. Aplikasi ini dinyatakan memenuhi indeks dari pengujian *usability* jika rata-rata presentase yang dihasilkan dari setiap kriteria lebih dari 60% sesuai pada Tabel 5. 12 Interpretasi Skor . Hasil perhitungan dari pengujian *usability* ditunjukkan pada Tabel 5. 13 Hasil Perhitungan Pengujian . Sedangkan hasil status pengujian *usability* ditunjukkan pada Tabel 5. 14 Hasil Status Pengujian .

Tabel 5. 12 Interpretasi Skor Likert

Skor Likert	Interpretasi Skor dengan Interval = 20	Pilihan
1	0%-19.99%	Sangat Tidak Setuju
2	20%-39.99%	Tidak Setuju
3	40%-59.99%	Netral
4	60%-79.99%	Setuju
5	80%-100%	Sangat Setuju

Keterangan:

Interval = 20 didapatkan dari hasil pembagian jumlah skor Likert dengan nilai 100

Tabel 5. 13 Hasil Perhitungan Pengujian *Usability*

No	Pernyataan	STS	TS	N	S	SS	Total Skor	Indeks (%)
1	Aplikasi ini cukup membantu saya dalam memutuskan lokasi mana yang akan menjadi tujuan saya	0	0	0	10	20	140	93.3%
2	Saya merasa aplikasi ini dapat membantu saya untuk mengetahui informasi lokasi yang akan saya tuju.	0	0	0	16	14	134	89.3%

3	Aplikasi ini bermanfaat untuk saya.	0	0	2	15	13	131	87.3%
Rata-Rata Indeks Daya Guna Aplikasi								90%
4	Saya tidak perlu mengeluarkan banyak tenaga (menekan tombol, menggeser layar) dalam menggunakan aplikasi ini	0	0	2	11	17	135	90%
5	Navigasi dalam aplikasi ini memudahkan saya dalam penggunaannya	0	0	3	12	15	132	88%
6	Saya merasa aplikasi ini sangat mudah untuk digunakan.	0	0	1	13	16	135	90%
Rata-Rata Indeks Kemudahan Penggunaan								89.3%
7	Saya dengan cepat dan mudah mempelajari aplikasi ini tanpa membaca panduan atau bertanya terlebih dahulu	0	1	2	11	16	132	88%
8	Saya dengan cepat mengingat bagaimana cara menggunakan aplikasi ini	0	0	0	12	18	138	92%
Rata-Rata Indeks Kemudahan Dalam Mempelajari Aplikasi								90%
9	Saya menyukai tampilan antarmuka pada aplikasi ini.	0	1	5	14	10	123	82%
10	Aplikasi ini berjalan sesuai dengan keinginan saya	0	0	6	11	13	127	84.7%
11	Saya merasa puas dengan aplikasi ini	0	0	1	16	13	132	88%
Rata-Rata Indeks Kepuasan								84.9%

Keterangan:

STS : Sangat Tidak Setuju

S : Setuju

TS : Tidak Setuju

SS : Sangat Setuju

N : Netral

$$\begin{aligned}
 \text{Total Skor} &= \Sigma \text{STS} \times 1 + \Sigma \text{TS} \times 2 + \Sigma \text{N} \times 3 + \Sigma \text{S} \times 4 + \Sigma \text{SS} \times 5 \\
 Y &= \text{Skor Likert tertinggi} \times \text{jumlah responden} \\
 &= 5 \times 30 \\
 &= 150 \\
 \text{Indeks (\%)} &= \frac{\text{Total Skor}}{Y} \times 100\% \quad (5-1)
 \end{aligned}$$

Tabel 5. 14 Hasil Status Pengujian *Usability*

Kriteria	Rata-Rata Indeks (%)	Status
Daya Guna Aplikasi (<i>Usability</i>)	90%	Sangat Setuju
Kemudahan Penggunaan (Ease of Use)	89.3%	Sangat Setuju
Kemudahan Dalam Mempelajari Aplikasi (Ease of Learning)	90%	Sangat Setuju
Kepuasan (Satisfaction)	84.9%	Sangat Setuju

Perhitungan presentase rata-rata dari semua kriteria

= jumlah total presentase (%) / jumlah kriteria

= $90 + 89.3 + 90 + 84.9 / 4$

= $354.2 / 4$

= 88.6 %

Dari hasil pengujian *usability* yang dilakukan menunjukkan bahwa aplikasi memenuhi empat kriteria yang ada dengan presentase rata-rata dari setiap kriteria sebesar 88.6% sehingga menunjukkan bahwa aplikasi rekomendasi lokasi WiFi memenuhi kriteria *usability*.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut:

1. Aplikasi rekomendasi lokasi WiFi dapat dibangun dengan dirancang dengan menggunakan diagram *use case*, *activity diagram*, *sequence diagram*, basis data, dan antarmuka. Rumus Haversine diimplementasikan dalam program untuk mendapatkan estimasi jarak antara pengguna dengan lokasi penyedia layanan Internet. Metode *weighted product* digunakan untuk memberikan penilaian tiap-tiap lokasi penyedia layanan Internet terhadap faktor jarak dari pengguna, biaya, dan kenyamanan tempat. Nilai yang diperoleh untuk setiap lokasi diurutkan secara menurun dari yang terbesar hingga yang terkecil untuk merepresentasikan urutan lokasi penyedia layanan Internet yang direkomendasikan oleh sistem dan ditampilkan dalam bentuk *list* di dalam aplikasi. Lokasi yang memiliki nilai tertinggi adalah lokasi yang paling direkomendasikan oleh sistem.
2. Hasil akurasi dari dari perbandingan perhitungan manual dan perhitungan oleh sistem sebesar 100%. Sehingga Implementasi metode *weigthed product* pada aplikasi berbasis Android dinyatakan akurat dikarenakan hasil dari perhitungan manual dan perhitungan yang dilakukan oleh sistem adalah sesuai.
3. Berdasarkan hasil uji validasi dengan metode *black box testing*, aplikasi dinyatakan telah memenuhi kebutuhan fungsional yang telah dianalisis. Berdasarkan hasil uji akurasi dari perbandingan pada proses perhitungan manual dan dengan proses perhitungan yang dilakukan oleh sistem, metode *weighted product* yang diterapkan pada aplikasi berbasis Android dinyatakan efektif..
4. Berdasarkan hasil uji *usability* yang dilakukan, terlihat bahwa aplikasi telah memenuhi seluruh faktor yang mempengaruhi *usability*, dalam hal *ease of*

use, ease of learning, dan *satisfaction* dengan indeks presentase rata-rata dari semua kriteria sebesar 88.6%.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi rekomendasi lokasi WiFi adalah sebagai berikut:

1. Untuk pengembangan lebih lanjut, perlu dilakukan penambahan kategori lokasi sehingga memudahkan pengguna untuk mencari rekomendasi dari kategori yang dimaksud.
2. Kriteria yang digunakan untuk perhitungan pada aplikasi ini akan lebih baik jika ditambahkan dengan kriteria lain seperti rating dari review pengguna sehingga keterlibatan pengguna dapat mempengaruhi rekomendasi lokasi yang ada.
3. Proses perhitungan lebih baik ditambahkan seperti *Analytical Hierarcy Proses* (AHP) untuk melakukan pembobotan kriteria sehingga proses pembobotan bisa lebih terstruktur dan nilai rekomendasi semakin akurat.
4. Untuk pengembangan aplikasi lebih lanjut, perlu dilakukan penambahan fitur untuk administrator agar lebih mudah untuk melakukan perubahan atau penambahan data.
5. Pengujian akurasi lebih baik dilakukan dengan menambahkan perbandingan antara hasil rekomendasi aplikasi dan pilihan pengguna untuk mendapatkan kecocokan hasil aplikasi.

DAFTAR PUSTAKA

- [BMC-14] MA, Ahmad, 13 Februari 2014, "Statistik: Internet, Sosial Media dan Mobile di Indonesia 2014", Bebmen, <http://bebmen.com/4027/statistik-Internet-sosial-media-dan-mobile-di-indonesia.html> (Diakses pada tanggal 1 Juni 2013)
- [DTI-09] Deteksi, 25 Juni 2009, "Kekurangan dan Kelebihan Modem USB", Deteksi, <http://deteksi.info/2009/06/kekurangan-dan-kelebihan-modem-usb/> (Diakses pada tanggal 2 Juni 2013)
- [ILA-14] Worldometer, 7 Biliion Worlds, 15 Juli 2014, "Indoensia Internet Penggunas", Internetlivestats, <http://www.Internetlivestats.com/Internet-penggunas/indonesia/> (Diakses pada tanggal 15 Juli 2014)
- [WAU-01] Deidu, Horace, 11 Maret 2013, "Where are the Android penggunas?", Asymco, <http://www.asymco.com/2013/03/11/where-are-the-Android-penggunas.> (Diakses pada tanggal 1 Juni 2014)
- [MAD-95] Yoon, Paul & Hwang, Ching-Lai. 1995. Multiple Attribute Decision Making : An Introduction. Sage Publications, California.
- [FMA-06] Kusumadewi, S & kawan. 2006. Fuzzy Multi-Attribute Decision Making. Graha Ilmu, Yogyakarta.
- [ANC-12] Bontoseoaji, 11 Oktober 2012, "SpeedTest Test Kecepatan Internetmu", Aplikanologi, <http://www.aplikanologi.com/produktivitas/speedtest-test-kecepatan-bandwidth-Internetmu/> (Diakses pada tanggal 29 Oktober 2014)
- [ADC-14] Adi, 5 April 2014, "Rumus Haversine Di MySQL", adi-hidayat, <http://adi-hidayat.com/rumus-Haversine-di-mysql/> (Diakses pada tanggal 30 Oktober 2014)
- [CTE-13] Compiletimeerror, 21 Januari 2013, "Android : Why, When and How to use AsyncTask with example", http://www.compiletimeerror.com/2013/01/why-and-how-to-use-AsyncTask.html#.VFJ0W_mUcuc (Diakses pada tanggal 30 Oktober 2014)

- [ADR-10] Helmi, Syafrizal & Kawan. 2010. Analisis Data Untuk Riset Manajemen Bisnis. USU Press, Medan.
- [CMS-10] Peacock, Marisa, 13 April 2010, "The What, Why and How of Usability Testing", <http://www.cmswire.com/cms/web-engagement/the-what-why-and-how-of-Usability-testing-007152.php> (Diakses pada tanggal 6 November 2014)
- [PSP-12] Deviyanti, Geotiffany. 2012. Pembangunan Sistem Pendukung Keputusan Pemilihan Hotel di Yogyakarta dengan Metode *weighted product* Berbasis Web. UAJY. Yogyakarta.
- [SMT-14] Nugroho, Robertus, Dian, Anggita, & Lie, Yulius. 2014. *Simulation Modeling of Tabetl PCs Selection Using Weighted-Product Algorithm*. Information System Project, School of Information System. Bina Nusantara University, Jakarta.
- [MUC-14] Sauro, Jeff. 28 Mei 2014. 7 Ways to Find Users for Usability Testing. Measuringu. <https://www.measuringu.com/blog/finding-users.php> (Diakses pada tanggal 20 November 2014)
- [STC-01] Lund, Arnold. Measuring Usability with The USE Questionnaire. *STC Usability SIG Newsletter* Vol. 8 No. 2. Oktober 2001.
- [NNG-12] Nielsen, Jakob. 10 April 2012. "Mobile Site vs. Full Site", <http://www.nngroup.com/articles/mobile-site-vs-full-site/> (Diakses pada tanggal 17 Desember 2014)

LAMPIRAN

Lampiran 1. Kuesioner Awal Penelitian

KUESIONER PENENTUAN FAKTOR DALAM MEMILIH
TEMPAT WIFI DI TEMPAT UMUM

NAMA : _____

UNIVERSITAS : _____

1. Apakah anda pernah menggunakan fasilitas Internet di tempat umum (café, universitas, alun-alun, dll.) ?
 - a) Ya
 - b) Tidak

2. Mengapa anda menggunakan fasilitas tersebut?
 - a) Paket Internet mahal
 - b) Tidak memiliki Internet di tempat tinggal
 - c) Sinyal Internet modem tidak stabil
 - d) Lain-lain

3. Untuk apa anda menggunakan fasilitas Internet di tempat umum? (Jawaban minimal 1)
 - a) Streaming
 - b) Download
 - c) Browsing

4. Dimana tempat yang paling sering anda kunjungi ketika menggunakan fasilitas Internet di tempat umum?
 - a) Café
 - b) Universitas
 - c) Taman / Alun-Alun
 - d) Perpustakaan
 - e) Lain-lain

5. Dimana tempat lain yang paling sering anda kunjungi ketika menggunakan fasilitas Internet di tempat umum?
- Café
 - Universitas
 - Taman / Alun-Alun
 - Perpustakaan
 - Lain-lain

6. Apakah faktor kenyamanan menjadi alasan anda dalam memilih tempat tersebut?
- Ya
 - Tidak

7. Jika iya, manakah diantara faktor kenyamanan berikut yang menjadi pertimbangan anda dalam memilih tempat berInternet yang anda pilih? (Pilih minimal 1 jawaban)
- Adanya *smoking room* / dilarang merokok
 - Adanya tempat ibadah dan kelengkapannya
 - Adanya kamar mandi dan kebersihannya
 - Jumlah stop kontak
 - Terdapat AC
 - Terdapat *live music* / Musik
 - Pelayanan yang ramah
 - Kebersihan tempat
 - Suasana dan tata ruang

8. Apakah faktor *cost* yang dikeluarkan menjadi alasan anda dalam memilih tempat tersebut?
- Ya
 - Tidak

9. Jika iya, berapa rentang *cost* maksimal yang menjadi pertimbangan anda dalam memilih tempat tersebut?

Rp.

10. Apakah faktor jarak dari tempat anda berada menjadi alasan anda dalam memilih tempat tersebut?
- Ya
 - Tidak

11. Jika iya, berapa jarak maksimal yang menjadi pertimbangan anda dalam memilih tempat tersebut?

Meter

12. Apakah faktor kecepatan Internet menjadi alasan anda dalam memilih tempat tersebut?
- Ya
 - Tidak

13. Dari keempat kriteria diatas (kenyamanan, kecepatan Internet, *cost* yang dikeluarkan, dan jarak dari tempat anda berada) manakah yang menjadi urutan prioritas anda dalam memilih tempat berInternet?

- Kenyamanan
- Kecepatan Internet
- Cost yang dikeluarkan
- Jarak dari tempat anda berada

Lampiran 3. Lembar Kuesioner Usability Testing

Nama : _____

Universitas : _____

--Kuesioner--

Berilah tanda silang (X) pada kolom jawaban yang menurut anda paling sesuai !

SS = Sangat Setuju

S = Setuju

N = Netral

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Pernyataan	STS	TS	N	S	SS
Daya Guna Aplikasi						
1	Aplikasi ini cukup membantu saya dalam memutuskan lokasi mana yang akan menjadi tujuan saya					
2	Saya merasa aplikasi ini dapat membantu saya untuk mengetahui informasi lokasi yang akan saya tuju.					
3	Aplikasi ini bermanfaat untuk saya.					
Kemudahan Penggunaan						
4	Saya tidak perlu mengeluarkan banyak tenaga (menekan tombol, menggeser layar) dalam menggunakan aplikasi ini					
5	Navigasi dalam aplikasi ini memudahkan saya dalam penggunaannya					
6	Saya merasa aplikasi ini sangat mudah untuk digunakan.					
Kemudahan Dalam Mempelajari Aplikasi						
7	Saya dengan cepat dan mudah mempelajari aplikasi ini tanpa membaca panduan atau bertanya terlebih dahulu					
8	Saya dengan cepat mengingat bagaimana cara menggunakan aplikasi ini					
Kepuasan						
9	Saya menyukai tampilan antarmuka pada aplikasi ini.					
10	Aplikasi ini berjalan sesuai dengan keinginan saya					
11	Saya merasa puas dengan aplikasi ini					

Lampiran 3. Data Riil Aplikasi

id	nama	alamat	skor_kenyamanan	kecepatan_Internet	biaya	longitude	latitude	informasi
1	McDonalds	Jl. MT. Haryono 115	333	52	6000	112.6128258 7052564	-7.946768833808532	Berkawasan di daerah sekitar Universitas Brawijaya, McD satu ini menjadi kawasan favorit bagi para mahasiswa. Hanya dengan membeli McFlurry seharga 7000, pengunjung bisa menikmati akses WiFi yang telah tersedia. Tempat ini memiliki fasilitas yang cukup lengkap yaitu adanya kamar mandi, tempat ibadah, dan kawasan bebas asap rokok. Kendati begitu, stop kontak hanya terdapat di spot-spot tertentu.

2	Mays	Jl. Mayjend M.T. Haryono No 28	359	127	8000	112.612074	-7.945098	<p>Mays Coffee (Brownies & Bakery) terletak di sebuah ruko di sekitar Mayjend M.T. Haryono. Kafe bertingkat dua ini tidak sulit ditemukan karena berada di pinggir jalan besar. Tata ruang yang bagus dan bersih serta dekorasi berupa foto-foto kota malang pada jaman dulu dipadu dengan harga menu yang relatif murah membuat pengunjung betah berlama-lama. Dengan rentang harga antara Rp 8000 - Rp 20000 pengunjung bisa menikmati menu yang ada. Tidak perlu khawatir jika ingin menggunakan WiFi, karena kafe ini menyediakan jaringan</p>
---	------	--	-----	-----	------	------------	-----------	--

								WiFi kencang dan stop kontak yang memadai.
3	Nyit Nyot Kafe	Jl. Terusan Surabaya	209	57	8000	112.615952	-7.964863	NyitNyot Kafe memiliki tempat yang relatif kecil namun nyaman. Disertai akses Internet yang lumayan kencang, kafe yang berada di area dekat Universitas Negeri Malang ini memiliki berbagai macam menu dengan rentang harga Rp 6000-Rp 12000. Terbatasnya stok kontak yang ada tidak menghilangkan suasana cozy yang dimiliki kafe kecil ini. Pelayanan yang ramah, akses yang relatif mudah, dan interior yang khas merupakan nilai tambah kafe ini.

4	LegiPait	Jl. Patimura 24	156	158	8000	112.634665	-7.973681	Legipait merupakan tempat ngopi yang resmi dibuka pada bulan Juni 2011. Legipait ini menempati bangunan tua kecil dengan interior bertema jadul (jaman dulu) yang terkesan menarik. Dengan harga kisaran Rp 8000-Rp 12000 anda dapat menikmati menu yang ada dan fasilitas jaringan WiFi yang kencang. Tak perlu khawatir dengan ketersediaan stop kontak karena tersedia disetiap meja yang ada. Tak jarang kafe satu ini penuh sesak oleh pengunjung.
5	Kedai Kopi NKRI	Jl. Besar Ijen 91	156	44	6000	112.625331	-7.963132	Kedai Kopi NKRI merupakan kedai kopi yang berlokasi di Jalan Besar Ijen. Tempat ini

								<p>cukup nyaman dan ramai pengunjung. Bermacam-macam racikan kopi nusantara yang tersedia serta keramahan penjual merupakan daya tarik tersendiri dari kedai kopi ini. Harga menu yang tersedia juga cukup murah. Cukup dengan budget antara Rp 6000 hingga Rp 8000, pengunjung bisa menikmati secangkir kopi yang nikmat serta jaringan Internet yang ada. Stop kontak yang tersedia pun cukup untuk tiap meja yang ada. Tak ayal kedai yang terletak di depan SOB ini diminati oleh pengunjung.</p>
6	Dunkin Donuts	Jl. Raya Tlogoma	340	128	8500	112.602884	-7.931575	Dunkin Donuts ini berada di kawasan pom bensin Jalan

	(Sengkaling)	s							<p>Raya Tlogomas. Lokasi ini tak jarang menjadi alternatif peristirahatan para pendatang sembari mengisi bensin. Tak jarang juga banyak orang-orang nongkrong karena tempatnya yang nyaman serta Internet yang kencang. Stop kontak yang ada pun mencukupi. Dengan harga donat satuan 8500, anda bisa menikmati tempat ini dan berselancar dengan kecepatan yang cukup kencang.</p>
7	Indomaret Point	Jl. MT Haryono	323	64	25000	112.612513	-7.9465		<p>Indomaret point memberikan layanan berupa full speed WiFi dari pembelian minimal Rp 25000. Fasilitas yang ada berupa stop kontak, smoking room, dan kamar mandi.</p>

								Tempat ini merupakan tempat favorit untuk berlama-lama nongkrong sembari melakukan kegiatan berInternet atau sekedar nonton bareng.
8	Republika	Jl. MT Haryono Gg XI	156	125.3	12000	112.610179	-7.94263	Republika kafe merupakan kafe yang berlokasi di daerah MT. Haryono. Aksesnya cukup mudah. Tempat yang nyaman, pelayanan yang ramah, serta interior yang unik menjadi daya tarik kafe ini. Dengan harga kisaran Rp 12000-Rp 29000, pengunjung bisa menikmati menu yang nikmat serta Internet yang kencang.
9	8oz	Jl. Citandui	325	48	16000	112.643341	-7.956997	8oz Kafe Studio merupakan sebuah kafe yang terletak di

		74						Jalan Cintandui. Kafe ini menyediakan tempat yang cukup nyaman dengan adanya smoking room, AC, serta suasana yang khas sehingga membuat pengunjung betah berlama-lama. Dengan biaya sekitar Rp 16000 - Rp 30000 pengunjung bisa menikmati menu-menu yang tersedia dan akses WiFi yang relatif kencang.
10	Dunkin Donuts (Suhat)	Jl. Sukarno Hatta	277	112	8500	112.619038	-7.944801	Dunkin Donuts ini berada di Jalan Sukarno Hatta. Lokasi ini cukup mudah ditemui karena berada di pinggir jalan besar serta dikelilingi oleh tempat makan dan kafe yang cukup dikenal di kota Malang. Pemandangan jalanan suhat

								<p>yang indah pada malam hari di lantai dua menjadi favorit pengunjung. Stop kontak yang ada pun mencukupi. Dengan harga donat satuan Rp 8500, anda bisa menikmati tempat ini dan berselancar dengan kecepatan yang cukup kencang.</p>
--	--	--	--	--	--	--	--	--