

Aplikasi Rekomendasi Lokasi Pariwisata Menggunakan Metode Profile Matching

Bayu Adhitia Wardana^{*1}, Zamah Sari², Denar Regata Akbi³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

bayu.adhitia@gmail.com^{*1}, abdzamahsari@gmail.com², dnarregata@umm.ac.id³

Abstrak

Lokasi Pariwisata atau objek wisata adalah sebuah tempat rekreasi atau tempat untuk berwisata. Lokasi Pariwisata dapat berupa objek wisata alam seperti gunung, danau, sungai, pantai dan juga bisa berupa bangunan seperti museum, benteng dan situs peninggalan sejarah. Merancang dan merealisasikan serta menguji aplikasi Android yang dapat memberikan rekomendasi lokasi pariwisata serta panduan wisata Kota Banjarmasin dengan menggunakan metode Profile Matching untuk mencari atau mendapatkan rute terpendek dalam radius tertentu yang akan ditampilkan pada Google Map API. Tujuan dari penelitian ini adalah mengimplementasikan Metode Profile Matching untuk mencari atau mendapatkan rute terpendek dalam radius tertentu yang akan ditampilkan pada Google Map API. Tujuan dari penelitian ini adalah mengimplementasikan Metode Profile Matching untuk bisa merekomendasi lokasi pariwisata dalam radius yang telah ditentukan oleh pengguna sehingga pengguna dapat lebih efektif dan cepat dalam mencapai ke tempat lokasi pariwisata yang telah ditentukan dan pengguna bisa memperkirakan bagaimana pengguna akan menuju ke lokasi pariwisata tersebut. Dalam aplikasi juga telah disediakan berupa petunjuk arah menuju ke lokasi pariwisata yang telah dipilih pengguna menggunakan Google Map API beserta rute alternative jika terjadi sesuatu yang membutuhkan pengalihan rute.

Kata Kunci: Lokasi Pariwisata, Google Map API, Profile Matching

Abstract

The tourism location or tourist attraction is a recreation place or a place to travel. Tourism location can be a natural attraction such as mountains, lakes, rivers, beaches and also can be buildings such as museums, forts and historical heritage sites. Designing and realizing and testing Android applications that can provide recommendations of tourism locations and travel guides of Banjarmasin City by using the Profile Matching method to find or get the shortest route in the radius will be displayed on the Google Map API. The purpose of this research is to implement the Profile Matching method to recommend tourism in a radius that has been determined by the user so that the user can be more effective and fast in reaching the location Tourism that has been defined and the user can be able to identify how the user will be heading to the location of the tourism. In the application also has been provided in the form of directions to the tourism site that has been selected by users using the Google Map API along with alternative routes if something happens that requires switching route.

Keywords: Tourism Location, Google Map API, Profile Matching

1. Pendahuluan

Pada penelitian ini penulis akan merancang serta membangun sistem rekomendasi serta panduan wisata kota Banjarmasin yang ditujukan untuk wisatawan yang membutuhkan bantuan dalam pemilihan rujukan spot wisata dan ulasan informasi disekitarnya dengan menggunakan metode Profile Matching Metode Profile Matching digunakan untuk menentukan rekomendasi berdasarkan jarak, tarif atau tiket dan fasilitas pendukung yang disediakan atau terdapat disekitar objek wisata yang direkomendasikan

1.1 Profile Matching

Pengertian Profile Matching dalam penentuan kenaikan suatu jabatan dapat dijelaskan sebagai proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya atau yang disebut juga gap, semakin kecil

gap yang dihasilkan maka bobot nilainya semakin besar berarti memiliki peluang lebih besar untuk karyawan menempati posisi tersebut. *Profile Matching* merupakan suatu proses yang sangat penting dalam manajemen SDM di mana terlebih dahulu ditentukan kompetensi atau kemampuan dasar yang diperlukan oleh suatu jabatan. Kompetensi kemampuan tersebut haruslah dapat dipenuhi oleh pemegang atau calon yang akan dinilai kinerjanya [1].

1.2 Google Map API

Google Map Service adalah sebuah layanan peta global virtual gratis dan *online* yang disediakan oleh perusahaan *Google*. *Google Maps* menawarkan peta yang diperoleh dari gambar satelit untuk pemetaan permukaan dan jalur di seluruh dunia. *Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan [2]. *Google Maps Application Programming Interface* adalah sebuah layanan yang diberikan oleh *Google* kepada para pengguna dan pengembang aplikasi untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi yang di dalamnya membutuhkan panduan rute dan pencarian lokasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta dan menambah konten melalui berbagai jenis layanan yang dimiliki serta mengizinkan kepada pengembang untuk membangun aplikasi. Pengguna dapat memanfaatkan layanan-layanan yang ditawarkan oleh *Google Maps* setelah melakukan registrasi dan mendapatkan *Google Maps API Key*. *Google* menyediakan layanan ini secara gratis kepada pengguna di seluruh dunia [2].

1.3 Firebase

Firebase merupakan suatu layanan *Database As a Service* yang disediakan oleh *Google* sebagai sarana pengembang aplikasi yang membutuhkan *database realtime*. *Firebase* merupakan platform untuk aplikasi *realtime*. Ketika data berubah, maka aplikasi dengan *firebase* akan meng-*update* secara langsung melalui setiap *device* perangkat baik web atau mobile. *Firebase* mempunyai *library* yang lengkap untuk sebagian besar platform web dan mobile dan dapat digabungkan dengan berbagai *framework* seperti *Java Script*, *AngularJS*, dan lain-lain. *Application Programming Interface (API)* untuk menyimpan dan sinkronisasi data akan disimpan sebagai susunan data dalam bentuk *JSON* pada *cloud* dan akan disinkronisasi secara *realtime*. Layanan pada *Firebase* meliputi autentikasi pengguna, pengaturan keamanan, dan *hosting*. Perubahan data pada satu *client* akan disinkronisasi pada semua *client* yang terdaftar ke data tersebut dalam hitungan milidetik. Kelebihan dari *Firebase* adalah dapat menerima data dari 1 juta perangkat secara bersamaan.

1.4 Aplikasi Android dan MIT APP Inventor

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi. *Android* diluncurkan untuk umum pada musim gugur di tahun 2008. *Android* sangat berkembang pesat di industri karena dua aspek utamanya yaitu bersifat *opensource* dan model arsitekturnya. Sebagai sebuah proyek yang bersifat *opensource*, memungkinkan *android* untuk sepenuhnya dipahami dan dianalisis mengenai fitur, penyelesaian pada bug program hingga hardware.

MIT APP Inventor merupakan model baru dalam pemrograman *android*, dimana tidak lagi menggunakan penulisan *source code* seperti pada umumnya, melainkan dengan menggunakan *puzzle drag and drop* yang memudahkan pengguna dalam membangun aplikasi *android*.

2. Metode Penelitian

Metodologi merupakan tahapan yang dilakukan oleh peneliti. Tahapan ini dimulai dari perumusan masalah hingga kesimpulan, yang nantinya akan membentuk sebuah alur yang sistematis yang akan berfungsi sebagai acuan penelitian agar hasil yang didapatkan tidak menyimpang dari tujuan yang telah ditetapkan. Pada penelitian ini metode penelitian yang ditempuh dimulai dengan analisa kebutuhan, desain sistem, pembuatan program, testing dan pengujian.

2.1 Pengambilan Data

Pada tahap ini akan mengidentifikasi permasalahan yang terjadi dan dialami oleh pengelola objek wisata di Kota Banjarmasin sehingga diperoleh data yang valid mengenai keluhan ataupun saran agar sistem ini dapat berfungsi optimal. Dengan mengidentifikasi

masalah secara terarah maka penelitian yang dilakukan akan memperoleh hasil yang sesuai dengan tujuan yang telah dipaparkan.

Permasalahan yang terjadi adalah terkadang objek wisata tertentu tidak terindex oleh *search engine* dan kurang mendapat perhatian dari wisatawan dikarenakan minimnya informasi, selain itu untuk membuat suatu objek wisata menjadi lebih menarik tentunya dibutuhkan informasi tambahan seperti akses menuju objek tersebut, ketersediaan fasilitas pendukung seperti hotel dan resto serta pusat oleh-oleh atau souvenir disekitar *spot* wisata dan rujukan kuliner.

2.2 Pemodelan data

Proses pengumpulan data dilakukan untuk mengetahui parameter apa saja yang didapat digunakan dalam penelitian ini, data yang akan digali oleh peneliti adalah tarif yang ditawarkan dari *spot* wisata, fasilitas pendukung, jarak ke pusat kota Banjarmasin serta koordinat dari tempat tersebut dan data ini dapat diperoleh dengan melakukan wawancara, survey secara langsung ke *spot* wisata. Data yang diperoleh dapat digunakan sebagai acuan membuat inputan pada bagian penyimpanan data [2] dan dibagi menjadi dua bagian yaitu *core factor* dan *secondary factor*.

Tabel 1. Tabel Penentuan Skor

No	Skor	Keterangan
1	5	sangat
2	4	cukup
3	3	sedang
4	2	kurang
5	1	tidak

Tabel 2. Contoh Model Alternatif dan Kriteria

No	Alternatif (objek wisata)	Kriteria	Skor
1	Penyedia jasa A	jarak dari pengguna aplikasi (semakin dekat skornya semakin tinggi)	2
		kisaran harga tiket atau tarif (semakin tinggi skornya semakin mahal)	5
		fasilitas disekitar (semakin tinggi skornya semakin banyak fasilitas serta layanan yang disediakan)	5
		ketersediaan objek pendukung (semakin tinggi skornya semakin banyak objek wisata yang berkaitan)	4
		akses jalan dan angkutan (semakin tinggi skornya semakin banyak)	3

Data yang diperoleh berupa:

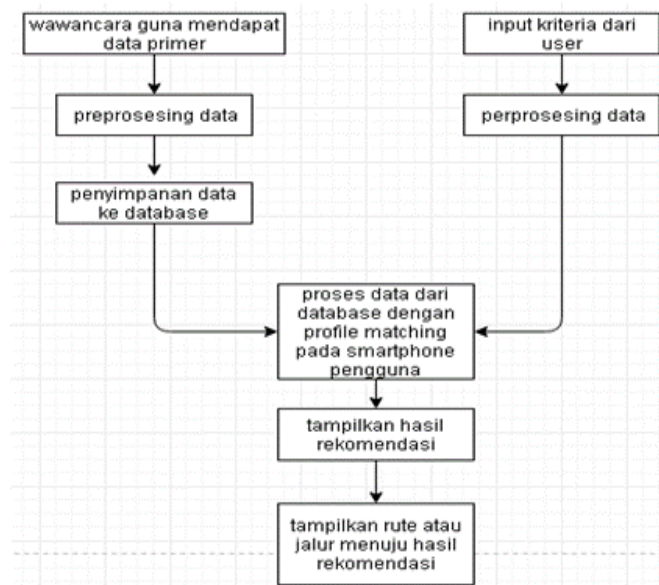
1. Data primer yang diperoleh dari pengamatan dan pengumpulan data dari media sosial dan website penyedia ulasan wisata.
2. Data sekunder merupakan data yang dihasilkan dari pemrosesan aplikasi berupa rekomendasi objek yang telah dijelaskan di pendahuluan.
3. Pada bagian di atas yang digunakan sebagai *core factor* adalah jarak dari pengguna aplikasi, harga tiket, fasilitas disekitar dan ketersediaan fasilitas pendukung, *secondary factor* yang digunakan adalah akses jalan.

2.3 Flowchart Program

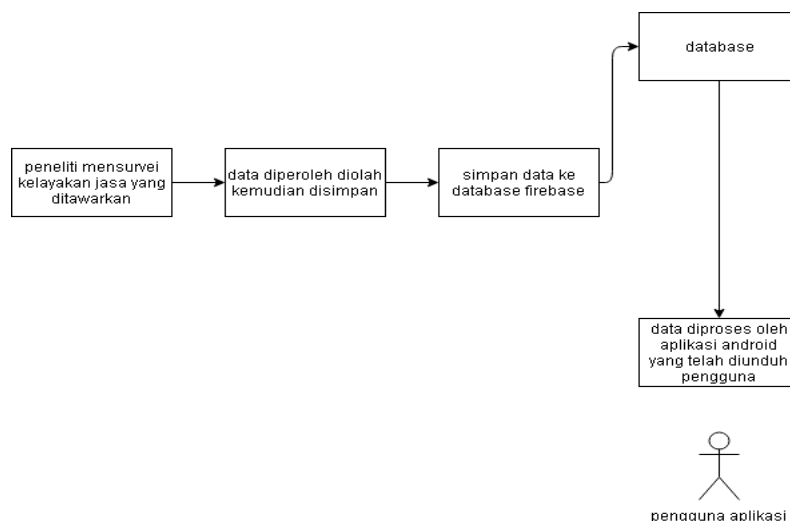
Pada tahap ini akan dijelaskan alur sistem yang di tempuh guna menjawab tujuan penelitian dan agar tidak menyimpang dari tujuan maka perlu didefinisikan perencanaan yang jelas. *Rule Based System* (basis pengetahuan) merupakan elemen utama dari metode *Profile Matching* untuk memberikan hasil yang maksimal.

Pada gambar 1 dijelaskan langkah-langkah yang dilalui oleh penulis sehingga penelitian dapat dikerjakan, dimulai dari pengumpulan data kemudian data tersebut diubah kedalam bentuk yang dapat diproses menggunakan *Profile Matching* untuk kemudian disimpan di *database*, dari database tersebut program android akan mengambil data untuk diproses dengan metode *Profile Matching*.

Profile Matching merupakan sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh individu, bukan tingkat minimal yang harus dipenuhi atau dilewati. Dalam *profile matching* dilakukan identifikasi terhadap suatu kelompok yang baik maupun buruk. Para individu dalam kelompok tersebut diukur menggunakan beberapa kriteria penilaian [4]. Kriteria rekomendasi objek wisata yang maksud antara lain adalah Jarak, tarif atau harga tiket yang ditawarkan, fasilitas pendukung dan akses menuju objek. Dari masing-masing kriteria ditentukan bobot dengan total bobot dari keseluruhan kriteria adalah 5, bobot yang digunakan berdasarkan tingkat kepentingan dalam pemberian rekomendasi.



Gambar 1. Alur Metode Perancangan



Gambar 2. Blok Diagram Sistem

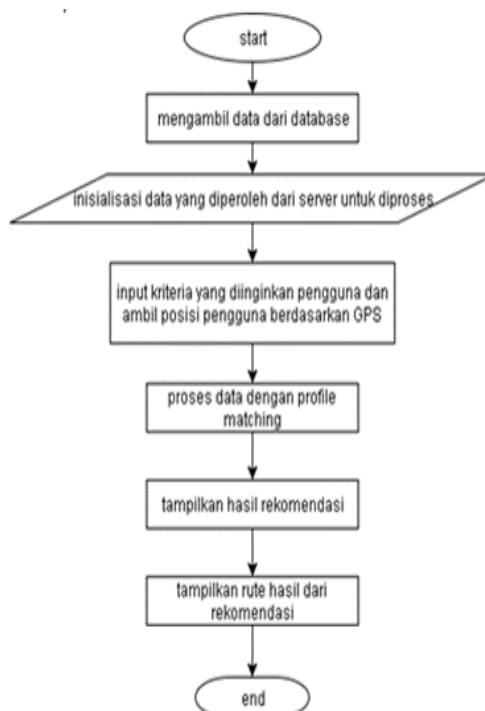
Pada Gambar 2 dijelaskan bahwa aplikasi ini menyimpan data objek dan segala informasi fasilitas pendukung yang terdaftar dalam sistem aplikasi, data tersebut diakses oleh aplikasi pada *smartphone* pengguna untuk memperoleh data rekomendasi yang diproses menggunakan metode *Profile Matching* sehingga dioutputkan suatu hasil rekomendasi kepada pengguna.

Proses perhitungan pada metode *Profile Matching*, diawali dengan pendefinisian nilai minimum untuk setiap variabel-variabel penilaian. Selisih setiap nilai data testing terhadap nilai minimum masing-masing variabel, merupakan *gap* yang kemudian diberi bobot. Bobot setiap variabel akan dihitung rata-rata berdasarkan kelompok variabel *Core Factor* dan *Secondary*

Factor. Komposisi *Core Factor* ditambah *Secndar Factor* adalah 100%, tergantung dari kepentingan pengguna metode ini. Tahap terakhir dari metode ini, adalah proses akumulasi nilai *Core Factor* dan *Secndar Factor* berdasarkan nilai-nilai variabel data *testing*.

Pembobotan pada metode *Profile Matching*, merupakan nilai pasti yang tegas pada nilai tertentu karena nilai-nilai yang ada merupakan anggota himpunan tegas atau *Crisp Set*. Di dalam himpunan tegas, keanggotaan suatu unsur di dalam himpunan dinyatakan secara tegas, apakah objek tersebut anggota himpunan atau bukan dengan menggunakan fungsi karakteristik. Langkah-langkah metode *profile matching* adalah:

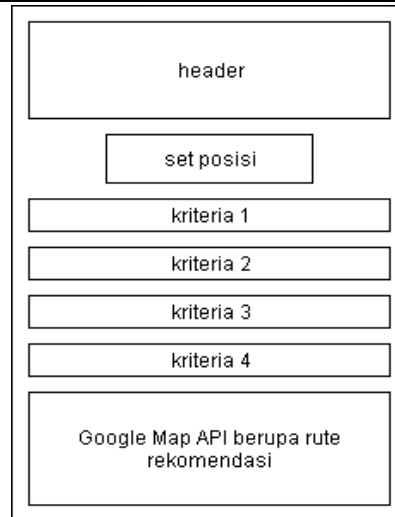
1. Menentukan variabel data-data yang dibutuhkan.
2. Menentukan aspek-aspek yang digunakan untuk penilaian.
3. Pemetaan *Gap profil*.
4. $Gap = \text{Profil Minimal} - \text{Profil data tes}$
5. Setelah diperoleh nilai *Gap* selanjutnya diberikan bobot untuk masing-masing nilai *Gap*. Perhitungan dan pengelompokan *Core Factor* dan *Secondary Factor*. Setelah menentukan bobot nilai *gap*, kemudian dikelompokkan menjadi 2 kelompok yaitu:
 1. *Core Factor* atau Faktor Utama, yaitu merupakan kriteria atau kompetensi yang paling penting atau menonjol atau paling dibutuhkan oleh suatu penilaian yang diharapkan dapat memperoleh hasil yang optimal.
 2. *Secondary Factor* atau faktor pendukung, yaitu merupakan item-item selain yang ada pada *core factor*, atau dengan kata lain merupakan faktor pendukung yang kurang dibutuhkan oleh suatu penilaian.
6. Perhitungan nilai total. Nilai Total diperoleh dari prosentase *core factor* dan *secondary factor* yang diperkirakan berpengaruh terhadap hasil tiap-tiap profil.
7. Perhitungan penentuan ranking. Hasil Akhir dari proses *profile matching* adalah *ranking*. Penentuan ranking mengacu pada hasil perhitungan tertentu.



Gambar 3. Flowchart Program

Pada Gambar 3 dijelaskan alur program jika akan menanyakan kriteria yang dikehendaki oleh pengguna aplikasi ini sehingga dapat disesuaikan ketika pemrosesan menggunakan metode *Profile*

Matching, kemudian hasil dari *Profile Matching* adalah berupa rekomendasi tempat menginap beserta koordinatnya yang kemudian ditampilkan menggunakan *google MAP API* sehingga diperoleh rute terdekat dari posisi pengguna aplikasi ini.



Gambar 4. Bentuk Awal Aplikasi

Pada Gambar 4 digambarkan bentuk awal aplikasi yang digunakan oleh *user* aplikasi pada aplikasi tersebut terdapat *slider* yang digunakan oleh pengguna untuk memasukkan kriteria yang diinginkan dan setelah kriteria tersebut terisi maka aplikasi akan meminta data dari server untuk kemudian diproses dengan *Profile Matching* sesuai dengan input kriteria dari pengguna, setelah proses *Profile Matching* selesai maka program akan mengeluarkan jalur menuju objek dan ulasan mengenai fasilitas di sekitar objek yang direkomendasikan.

2.3 Pembuatan Program

Tahap ini merupakan proses penulisan program yang merupakan penerjemahan dari bentuk design ke dalam bahasa yang bisa dikenali oleh komputer. Proses ini dilakukan oleh programmer yang akan menerjemahkan data wawancara dan penelitian yang diminta oleh *User*. Tahapan inilah yang merupakan tahapan nyata dalam mengerjakan suatu sistem. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

2.4 Testing

Pada tahap ini adalah tahap dimana untuk menguji hasil dari implementasi yang telah dilakukan apakah telah sesuai dengan keinginan atau belum. Kemudian mengevaluasi hasil implementasi agar didapatkan hasil yang maksimal dari aplikasi ini. Untuk *testing* aplikasi ini akan menggunakan metode *Black Box* guna mengetahui keseluruhan fungsi telah bekerja dengan baik, dalam artian fitur-fitur yang dicantumkan pada program dapat digunakan.

2.5 Implementasi dan Pengujian

Setelah diimplementasikan berupa aplikasi *anroid* maka perlu dilakukan pengujian guna mengetahui sejauh mana kualitas dari rekomendasi yang dihasilkan serta seberapa tepat panduan rute yang ditampilkan. Analisis akan difokuskan pada rekomendasi dan rute yang dihasilkan oleh *Haversine Formula* yang diterapkan pada *Application Programming Interface Google Map* dengan mencocokkan antara hasil yang diharapkan pengguna dan hasil yang dikeluarkan.

Tabel 1. Skenario Pengujian

No	Model pengujian	Hasil	keterangan
1	mencoba mengukur jarak antara dua koordinat <i>latitude</i> dan <i>longitude</i> yang	Hasil berupa jarak terukur dalam satuan kilometer	deskripsi titik <i>latitude</i> dan <i>longitude</i> yang akan diukur serta hasil pengukuran

	diperoleh dari <i>google map</i> dengan jarak yang tidak terlalu jauh (100m) menggunakan metode <i>haversine formula</i> [2][3]. terkait dengan pengujian nomor 1 maka diperlukan uji dengan fungsi estimasi waktu tempuh yang disediakan oleh <i>google map</i> [2]. dilakukan pengujian pencarian rute terpendek dan waktu tempuh tercepat yang mana pada pengujian ini <i>google map API</i> telah menyediakan fasilitas untuk rute[2]. uji fungsi dan fitur-fitur aplikasi	Estimasi waktu tempuh dalam satuan jam, menit dan detik	mencoba melakukan perjalanan untuk mengukur waktu tempuh menggunakan <i>stopwatch</i>
2			
3		hasil berupa daftar rekomendasi rute dan waktu tempuh	deskripsi hasil pengujian berupa paragraf
4		hasil berupa deskripsi pengujian	dipaparkan dalam bentuk paragraf

3. Hasil dan Pengujian

3.1 Hasil Implementasi *Block Program*

Setelah tahap implementasi block program maka akan diperoleh hasil berupa *interface* yang tersusun dalam beberapa tampilan pada menu-menu aplikasi. Desain dari *user interface* yang baik dalam sebuah sistem dapat mempermudah *user* atau pengguna dalam menggunakan aplikasi tersebut. *Interface* pada aplikasi ini menggunakan bahasa pemrograman android *MIT APP Inventor*. Berikut tampilan *interface* pada sistem yang sudah dibangun.



Gambar 5. Tampilan Halaman Awal

Pada Gambar 5 merupakan tampilan ini adalah halaman depan yang pertama kali diakses oleh pengguna atau *user*. Terdapat keterangan lokasi dengan *Latitude* dan *Longitude*, tombol *Set Point* untuk mengambil posisi *user* saat ini, *slider-slider* untuk menerima input kriteria dari *user* dan tombol cari untuk memproses semua input yang diterima kedalam *profile matching* dan dikeluarkan sebagai suatu rekomendasi kepada *user*.



Gambar 6. Tampilan User Memasukan Kriteria

Pada Gambar 6 merupakan proses ketika *user* sedang memasukkan kriteria yang diinginkannya dengan inputan yang harus dipenuhi adalah jarak, harga tiket, akses dan fasilitas.

Pada Gambar 6 merupakan proses ketika *user* sedang memasukkan kriteria yang diinginkannya dengan inputan yang harus dipenuhi adalah jarak, harga dan fasilitas. Pada Gambar 6 *user* menginginkan jarak yang berkisar 4,4 km dari lokasinya saat ini, harga tiket paling minim, dengan fasilitas paling minim dan dapat diakses dengan kendaraan roda dua, Set point yang dihasilkan adalah *Latitude* -7.91984 dan *Longitude* 112.5941, ini merupakan posisi *user* saat ini yang akan dibandingkan dengan input jarak pada perhitungan *profile matching*.



Gambar 7. Hasil Rekomendasi

Pada Gambar 7 merupakan bentuk dari tampilan hasil rekomendasi, pada bagian ini *user* akan memperoleh jarak ke rekomendasi yang dihasilkan, untuk mengetahui rute yang ditempuh *user* cukup mengklik hasil rekomendasi yang telah diurutkan dari yang paling dekat ke terjauh dan akan memunculkan rute dalam format *Google map*.



Gambar 8. Rute Rekomendasi

Pada Gambar 8 merupakan bentuk dari tampilan rute dalam *Google map* yang dapat digunakan oleh *user* sebagai panduan menuju lokasi yang direkomendasikan, dengan menggunakan *Google map* ini akan mempermudah *user* karena telah disediakan panduan suara.

3.2 Pengujian Dengan Metode *Blackbox*

Pada tahap ini dilakukan pengujian fungsional dengan menggunakan *black box testing*. Pengujian ini dilakukan untuk mengetahui apakah aplikasi ini sudah sesuai rancangan dan kebutuhan atau masih diperlukan perubahan dan penyempurnaan. Pengujian ini difokuskan pada skenario pengujian aplikasi yang ditunjukkan pada Tabel 2 dengan tujuan untuk mengetahui fungsi-fungsi aplikasi berdasarkan urutan penggunaan dan untuk mengevaluasi fungsi dari setiap tombol yang diberikan, apakah sudah sesuai dengan perancangan.

Tabel 2. Skenario pengujian *BlackBox*

No	Test	Hasil Yang Diharapkan
1	Menjalankan program untuk menampilkan halaman awal aplikasi	Program dapat menampilkan halaman awal aplikasi
2	Memasukkan nilai kriteria dari user dengan slider, slider ini terdiri dari kriteria jarak, harga, fasilitas dan akses	Mampu melakukan input kriteria dengan menggeser slider pada tampilan aplikasi
3	Uji coba <i>User</i> Memasukkan <i>Set Point</i> , <i>set point</i> merupakan pengambilan koordinat <i>user</i> aplikasi berdasarkan data <i>GPS</i> yang digunakan untuk mengukur preferensi jarak dari <i>user</i> ke data-data koordinat tempat rekomendasi di <i>database</i>	Menampilkan proses <i>Set Point</i> dan memunculkan koordinat <i>GPS</i> berupa <i>latitude</i> dan <i>longitude user</i>
4	Menampilkan hasil rekomendasi setelah <i>user</i> menekan tombol "Cari"	Menampilkan hasil rekomendasi wisata
5	Menampilkan foto dan keterangan hasil rekomendasi setelah <i>user</i> menekan salah satu <i>list</i> hasil rekomendasi	Menampilkan foto dan keterangan hasil rekomendasi.
6	Menampilkan rute ke rekomendasi dengan menggunakan <i>Google Map</i> setelah <i>user</i> menekan tombol "rute"	Menampilkan rute ke hasil rekomendasi pada <i>Google Map</i>

Pada Tabel 2 merupakan urutan atau skenario pengujian *blackbox* aplikasi rekomendasi wisata di Banjarmasin dengan metode *profile matching* berbasis *android*, tujuan pengujian tersebut adalah untuk mengetahui hasil dari setiap fungsi yang disebutkan pada bagian perancangan. Skenario yang dipaparkan dalam Tabel 2 dimulai dari *user* membuka aplikasi sampai dengan *user* memperoleh rekomendasi dan menampilkan rute menuju rekomendasi.

4. Kesimpulan

Dilihat dari penelitian di atas dapat disimpulkan aplikasi yang dibangun akan bisa memberikan rekomendasi tempat pariwisata terdekat dari set point yang telah di set oleh user untuk mencari rute terdekat dan rute alternatif kedalam bahasa pemrograman berbasis *android*,

sehinggaimenghasilkan rekomendasiitempatpariwisataidalam radius tertentu yang ditampilkan pada *Google Map API*.

Referensi

- [1] Turban, Efraim, Jay E. Aronson, and Ting-Peng Liang. "Sistem Pendukung Keputusan dan Sistem Cerdas." *Terjemahan Dwi Prabantini. Yogyakarta: Andi* (2005).
- [2] Setiawan, Cahyo. "Pembangunan Aplikasi Traveling Guide Kota Bogor Menggunakan *Google MAP API*." (2014).
- [3] Hafsah, Mrs, Frans Richard Kodong, and Alain Julian. "Sistem Pendukung Keputusan Penentuan Hotel Dengan Menggunakan Metode Promitee dan AHP." *Seminar Nasional Informatika 2011*. 2011.
- [4] Chairi, A., Putri, R., & Fanani, L. 2017 Agu 30. Rekomendasi Tempat Wisata Kota Malang Menggunakan Metode Profile Matching Dan Saran Rute Menggunakan Floyd Warshall Berbasis Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. [Online] 2:5
- [5] Arisandi, Arisandi, Muh Ihsan Sarita, and S. Sagala La Ode Hasnuddin. "Sistem Penunjang Keputusan Pemilihan Hotel di Kota Kendari Berbasis Web Menggunakan Metode *Fuzzy Tahani*." *semanTIK 2.1* (2016).