PENENTUAN JALUR TERPENDEK MENUJU CAFE DI KOTA MALANG MENGGUNAKAN METODE BELLMAN-FORD DENGAN LOCATION BASED SERVICE BERBASIS ANDROID

M.Rofiq, Riza Fathul Uzzy
STMIK ASIA Malang

ABSTRAK

Dalam penelitian ini di lakukan penerapan perhitungan metode Bellman-Ford yang bertujuan untuk mencari jalur terpendek menuju cafe di kota Malang. Metode Bellman-Ford tersebut menghitung semua jalur dari tempat asal ke tempat tujuan yang terbentuk dalam suatu graf agar di temukan jalur terpendek berdasarkan data yang di dapat dari peta dan GPS, data-data tersebut yaitu jarak jalan, titik persimpangan jalan dan koordinat tempat asal dan tujuan. Dengan simulasi perhitungan dengan mengambil peta dari Open street map yang di dalamnya terdapat informasi jarak jalan, titik persimpangan jalan, koordinat tempat asal dan tujuan, dimana kampus Asia sebagai tempat asal dan cafe Kopi.Net sebagai tujuan sehingga terbentuk suatu graf. Metode Bellman-Ford ini menghitung jumlah jarak jalan antara tempat asal dengan beberapa persimpangan jalan yang akan dilaluinya pertama kali dengan nilai paling terkecil sehingga akan mengetahui jalan mana yang akan dipilih selanjutnya, dan persimpagan terpilih sebagai titik awal perhitungan yang berikutnya. proses perhitungan tersebut akan diulang sejumlah titik persimpangan yang ada sampai mendapatkan jumlah jarak jalan terpendek menuju tempat tujuan. Dari proses simulasi, bahwa metode Bellman-Ford bisa digunakan untuk menentukan jalur terpendek.

Kata Kunci : *Jalur Terpendek, Bellman-Ford, GPS, Café, peta*

ABSTRACT

In doing this research in the application of the calculation method of Bellman-Ford is aiming to find the shortest path towards the cafe in the city of Malang. Bellman-Ford method is to count all the lines from the place of origin to point of destination that is formed in a graph in order to find the shortest path based on data obtained from the map and GPS, these data are walking distance, the point of intersection and the coordinates of the place of origin and purpose. With simulation calculations to take a map of the Open street map in which there is information within the road, road intersection point, the coordinates of the place of origin and destination, where the Asian campus as a place of origin and destination cafe Kopi.Net as to form a graph. Bellman-Ford's method of calculating the amount of the distance between the place of origin with few crossroads that will be passed first to the most smallest value that will determine which path will be chosen later, and persimpagan selected as a starting point for the next calculation. The calculation process is repeated a number of the existing crossing point to get the amount within the shortest path to the destination. From the simulation process, that the Bellman-Ford method can be used to determine the shortest path.

Keywords: Shorthest path, Bellman-Ford, GPS, Café, map

PENDAHULUAN

Kota Malang sesuai dengan julukannya yaitu kota pendidikan tentu banyak sekali pendatang dari berbagai wilayah di indonesia yang singgah di Malang untuk mencari ilmu. Para pendatang tersebut kebanyakan adalah mahasiswa dan ada juga pegawai atau profesi lain tapi tidak sebanyak mahasiswa, kota Malang letaknya dekat dengan kota Batu yang terkenal dengan wisatanya, tentu banyak para pelancong yang berdatangan untuk melewati kota Malang atau singgah sejenek di kota Malang. Para pendatang tersebut tentu tidak mengetahui semua informasi yang ada di kota Malang lebih-lebih informasi tentang café yang berada di kota Malang. Walaupun banyak café yang terdapat di kota Malang, para pendatang tersebut kadang merasa kesulitan untuk mencari café yang cocok menurut mereka karena informasi tentang café yang berada di kota Malang sangat terbatas, meskipun sudah ada informasi yang disediakan oleh pemkot Malang yang memberikan

informasi nama café dan alamat café, hal tersebut tidak menjawab untuk mencari lokasi café lebih cepat karena tidak disertai peta dan penunjuk arah jarak terdekat menuju café, lebih-lebih informasi tersebut tidak di sertakan tentang fasilitas yang di tawarkan oleh café tersebut.

Pada saat ini teknologi GPS cukup membantu permasalahan pencarian lokasi. Kita bisa mudah menemukan lokasi yang kita tuju, akan tetapi informasi tentang café sendiri tidak tertera lengkap pada teknologi GPS tersebut. Untuk memberikan informasi lokasi café tersebut harus ada yang memberi tanda pada peta data yang terdapat pada GPS tersebut.

Solusi dari permasalahan tersebut bisa membuat suatu aplikasi pencarian jalur terpendek sekaligus dengan memberi informasi tentang café di kota Malang. Dalam pencarian jalur terpendek ada beberapa metode yang bisa di terapkan. metode seperti Djikstra, Bellman-Ford, Dan Floyd-Warshall adalah metode yang digunakan untuk pencarian jalur terpendek yang dapat digunakan untuk mencari lokasi café dan rute perjalanan menuju café tersebut.

Algoritma Dijkstra dapat digunakan untuk mencari rute terpendek dari suatu grafik berbobot. Algoritma Bellman-Ford fungsinya sama dengan Algoritma Dijkstra, hanya saja prosesnya memakan waktu yang lebih lama. Meskipun memakan waktu lebih lama, Algoritma Bellman-Ford dapat memberikan hasil yang lebih akurat. Algoritma Floyd-Warshall juga dapat menangani kasus serupa. Algoritma dijkstra lebih cepat dalam menentukan rute terpendek, namun untuk hasil yang lebih akurat, algoritma Bellman lebih disukai.

Android adalah sistem oprasi yang sangat populer pada saat ini karena banyak fitur-fitur dan aplikasi-aplikasi yang adapada Android. Sistem oprasi ini sering di digunakan pada perangkat mobile sehingga dapat digunakan dimanapun juga. Dengan kelebihan yang di milikinya android lebih banyak dipilih oleh masyarakat. Android adalah sistemoprasi yang *open source* sehingga dalam pengembangannya lebih cepat.

Dalam penulisan penelitian ini, ruang lingkup permasalahan dibatasi oleh beberapa hal, yaitu :

1. Hasil dari penelitian ini berupa sebuah aplikasi yang dikembangkan dengan menggunakan framework Android SDK dan berjalan pada smartphone dengan system operasi Android.
2. Menggunakan Metode Algoritma Bellman-Ford dan berbasis Location Based Service.
3. Studi kasus pembuatan aplikasi ini hanya untuk café yang terdapat di Kota Malang yang sesuai hasil observasi dan survei.
4. Aplikasi yang di hasilkan tidak mendukung real time way condition dan Multi user

LANDASAN TEORI

ANDROID

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

1. Fitur Android

a. Framework aplikasi, memungkinkan daur ulang dan penggantian komponen.
b. Browser terintegrasi berbasis engine Open Source WebKit yang juga digunakan di browser IPhone dan Nokia S60v3.
c. Rancangan handset. Platform disesuaikan dengan kebutuhan VGA (*Video Graphics Adapter*) yang lebih besar, library grafik 2D dan 3D yang berdasarkan pada spesifikasi OpenGL ES 1.0 serta layout smartphone yang tradisional.
d. Konektivitas. Android mendukung berbagai teknologi konektivitas seperti GSM (*Global System for Mobile Communications*) EDGE (*Enhanced Data rates for GSM Evolution*), CDMA (*Code Division Multiple Access*), EV-DO

(*Evolution-Data Optimized*), UMTS (*Universal Mobile Telecommunications System*), Bluetooth dan Wi-Fi (*Wireless Fidelity*).

e. Pesan. Android mendukung pengiriman pesan dalam bentuk SMS (*Short Message Service*) dan MMS (*Multimedia Messaging Service*).
f. Dukungan Java. Software yang ditulis dalam bahasa Java dapat dikompilasi dan akan dieksekusi pada mesin virtual Dalvik, yang merupakan implementasi dari VM (*Virtual Machine*) yang dirancang khusus untuk penggunaan perangkat bergerak.
g. Dukungan media. Android mendukung beberapa format audio/video seperti: H.263, H.264 (dalam kontainer 3GP atau MP4), MPEG-4 SP, AMR, AMR-WB (dalam kontainer 3GP), AAC, HE-AAC (dalam kontainer MP4 atau 3GP), MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF dan BMP.
h. Dukungan hardware tambahan. Android mendukung penggunaan kamera, layar sentuh, GPS (*Global Positioning System*), pengukur kecepatan, magnetometer, akselerasi 2D bit blits (dengan orientasi hardware, scaling, konversi format piksel) dan akselerasi grafis 3D.
i. Market. Mirip dengan App Store pada iPhone OS, Android Market adalah sebuah katalog aplikasi yang dapat di-download dan diinstal pada telepon seluler secara online, tanpa menggunakan PC (*Personal Computer*). Awalnya hanya aplikasi gratis saja yang didukung. Dan s ejak tanggal 19 Februari 2009 aplikasi berbayar telah tersedia di Android Market untuk Amerika Serikat.
j. Multi-touch. Android memiliki dukungan bawaan untuk multi-touch yang tersedia pada handset terbaru seperti HTC Hero. Pada awalnya fitur tersebut dinonaktifkan pada level kernel (mungkin untuk menghindari pelanggaran paten terhadap teknologi layar sentuh Apple). Sejak Google merilis update untuk Nexus One dan berencana juga untuk merilis update untuk Motorola Droid yang memungkinkan multi-touch.
k. Lingkungan pengembangan yang kaya, termasuk emulator, peralatan debugging, dan plugin untuk Eclipse IDE.

GLOBAL POSITIONING SYSTEM (GPS)

1. Pengertian GPS

Global Positioning System (GPS) merupakan sistem navigasi yang berbasiskan satelit dan merupakan alat untuk mengetahui posisi yang tersusun atas constellation 24 satellites yang mengorbit pada bumi pada ketinggian kurang lebih 11.000 mil. Awalnya GPS hanya terbatas untuk kalangan militer di USA, tetapi pada awal tahun 80an pemerintah membuatnya terbuka untuk digunakan secara umum khususnya pada komersial bisnis, travel, dan navigasi, sampai sekarang GPS sudah meluas penggunaannya seperti mendeteksi gempa dan ramalan cuaca. GPS didesain untuk beroperasi 24 jam, dalam segala kondisi cuaca dan bisa digunakan di seluruh dunia.

2. Cara Kerja GPS

Prinsip dasar dari GPS terletak pada jarak dari receiver ke satelit, receiver minimal harus mencari 3 posisi satelit untuk menghasilkan posisi yang akurat, operasi ini dinamakan triangulation, secara singkat triangulation dapat dijelaskan demikian ketiga satelit akan mencari irisan dari 3 posisi yang berbeda, posisi yang akurat akan ditemukan pada irisan ketiga satelit. Sebagai contohnya, misalkan kita disuruh oleh seseorang untuk menemukan seseorang (misalkan) di toko buku berdasarkan beberapa petunjuk yang diberikan oleh orang tersebut. Pertama, kita diberitahu bahwa kita tepat berada 10 miles jauhnya dari rumah kita. Kita akan mengetahui bahwa kita berada di suatu radius dengan jangkauan 10 miles. Dengan informasi ini, kita akan kesusahan mencarinya karena radiusnya sangat luas.

3. Location Based Service *(LBS)*

Menurut Qusay H. Mahmoud (J2ME and Location-Based Services, 2004), *Location Based Service* adalah sebuah layanan yang digunakan untuk mengetahui posisi dari pengguna, kemudian menggunakan informasi tersebut

untuk menyediakan jasa dan aplikasi yang personal.

Ada 2 pendekatan dasar yang dipakai dalam mengimplementasikan LBS, yaitu :

1. Memproses data lokasi diserver dan mengirimkan hasilnya ke alat.
2. Mendapatkan data lokasi dari alat tersebut berdasarkan aplikasi yang menggunakannya secara langsung.

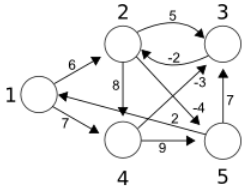
Aplikasi dapat menggunakan beberapa pendekatan yang disediakan, yaitu:

1. Menggunakan jaringan telepon seluler, akurasi dari metode ini tergantugn kepada ukuran dari sel dan mungkin tidak akurat.
2. Menggunakan satelit, dengan menggunakan bantuan GPS untuk mendapatkan data posisi yang akurat.

ALGORITMA BELLMAN-FORD

Algoritma Bellman-Ford adalah algoritma untuk menghitung jarak terpendek (dari satu sumber) pada sebuah digraf berbobot. Maksudnya dari satu sumber ialah bahwa ia menghitung semua jarak terpendek yang berawal dari satu titik node. Algoritma Dijkstra dapat lebih cepat mencari hal yang sama dengan syarat tidak ada sisi(edge) yang berbobot negatif . Maka Algoritma Bellman-Ford hanya digunakan jika ada sisi berbobot negatif.

Keunggulan lain yang membuat algoritma ini lebih baik dari algoritma lainnya yaitu algoritma ini memungkinkan bobot pada sisi yang menghubungkan antara dua simpul berupa bilangan negatif. Hal tersebut seperti dijelaskan oleh contoh di bawah ini.



Gambar Salah satu contoh graf dengan sisi bernilai negatif.

Dalam algoritma *Bellman-Ford*, apabila ingin dicari lintasan dengan bobot paling sedikit dari satu ke dua, maka lintasannya adalah 1-4-3-2, sehingga bobot yang didapat adalah 7 -3 -2 = 2.

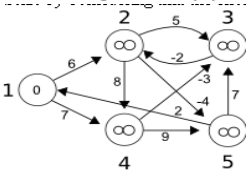
Berikut akan disajikan algoritma umum dari *Bellman-Ford* dalam notasi matematika.

$$M[i,v] = \min(M[i-1,v], (M[i-1,n] + C_{vn}))$$

i = iterasi, v = vertex = node, n = node neighbor, C = cost

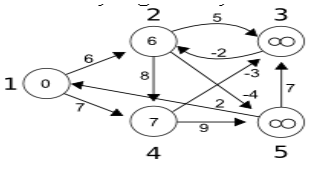
Sebelum memulai perhitungan dan penganalisaan, terlebih dahulu yang harus dilakukan adalah menamai setiap simpul dan memberikan bobot dari tiap sisi. Untuk sisi pada graf tak berarah harus mendefinisikannya sebanyak 2 kali, yakni dari titik pertama ke titik kedua dan sebaliknya dengan nilai yang sama. Namun, apabila yang akan diimplementasikan adalah suatu graf yang berarah maka cukup dengan mendefinisikannya sebanyak satu kali sesuai dengan arah graf.

Langkah pertama yang harus dilakukan dalam analisis graf menggunakan algoritma *Bellman-Ford* adalah menentukan titik asal setelah menetapkan titik asal dari lintasan, lalu melakukan penandaan simpul(*marking*). Dalam hal ini, semua titik yang bukan titik asal harus ditandai dengan infinity(∞). Titik asal sendiri, sebagai titik pangkal dari lintasan yang akan dibentuk, ditandai dengan nol (0).



Gambar Melakukan penandaan simpul.

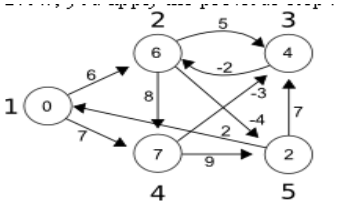
Selanjutnya melakukan *relaxing* pada simpul yang terdapat pada graf. Simpul yang di *relaxing* adalah simpul selain simpul asal. Berikut adalah gambar yang menjelaskan salah satu dari proses relaxing dari graf tersebut.



Gambar Relaxing tahap 1

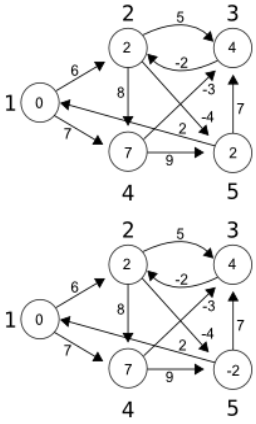
Relaxing disini berarti membandingkan bobot suatu titik, dalam hal ini telah ditandai dengan *infinity*, dengan titik lain yang berada disekitarnya yang menghubungkannya dengan titik asal. Dalam *relaxing* tahap 1 ditunjukkan

bahwa simpul 2 langsung diberikan bobot 6. Hal ini terjadi karena 6, besar bobot sisi yang menghubungkan antara simpul asal dengan simpul 2, lebih kecil daripada nilai sebelumnya, yakni *infinity*. Begitu pun yang terjadi pada simpul 4. simpul 4 diberikan bobot 7 karena bobot sisi yang menghubungkan simpul 4 dengan simpul asal adalah 7 yang dalam hal ini lebih kecil jika dibandingkan dengan nilai sebelumnya (*infinity*).



Gambar Relaxing tahap 2

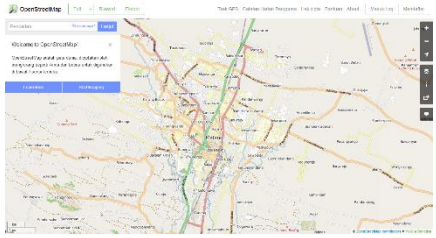
Dari gambar diatas, simpul 3 bernilai 4 karena bobot simpul yang berhubungan dengan simpul 3 ditambah bobot sisi yang menghubungkannya yang paling kecil berasal dari simpul 4 yang nilainya adalah 4 (hasil penjumlahan 7 -3). Simpul 5 bernilai 2 karena bobot hubungan terkecil yang dimilikinya adalah keterhubungan dengan simpul 2 yakni 2(hasil penjumlahan 6 – 4).



Gambar Relaxing tahap 3 dan tahap 4

Open Street Map

OpenStreetMap adalah sebuah alat untuk membuat dan berbagi informasi dalam bentuk peta. Siapapun dapat berkontribusi untuk OSM, dan ribuan orang menambahkan proyek setiap harinya. Para pengguna menggambarkan peta pada komputer, dibandingkan dengan kertas, tetapi kita akan melihat pada panduan ini, menggambar sebuah peta pada sebuah komputer tidak jauh berbeda dengan menggambar peta pada secarik kertas. Kita masih menggambarkan garis untuk merepresentasikan jalan, lapangan, dan lain-lain, dan kita masih merepresentasikan sekolah dan rumah sakit dengan simbol. Hal yang paling penting adalah peta OSM disimpan di dalam internet, dan siapapun dapat mengakses peta tersebut kapanpun, gratis.



Gambar Web Open Street Map

1. Bagaimana Osm Bekerja

OSM adalah gratis dan terbuka, memungkinkan untuk siapa pun mendownload semua data di dalam database ini. Namun, karena datanya terlalu besar (datanya lebih dari 30 GB bahkan ketika data tersebut dikompres), ini hampir tidak mungkin untuk bekerja dengan semua data sekaligus.

2. Menggunakan Geodata

Editor seperti iD atau JOSM memiliki satu fungsi utama bahwa mereka sangat baik dalam - membuat pengguna lebih mudah dalam mengedit OpenStreetMap. Tetapi mereka bukan perangkat lunak yang dimaksud untuk menganalisis atau meng-query data - fungsi ini sebaiknya diserahkan ke aplikasi lain. Perangkat lunak SIG yang gratis dan open source QGIS, memungkinkan pengguna untuk mendesain peta yang indah, meng-query dan menganalisis data, dan banyak lagi. Perangkat lunak SIG juga dapat digunakan untuk mengedit geodata, tetapi hal ini lebih mudah untuk mengedit OpenStreet dengan menggunakan editor OSM.

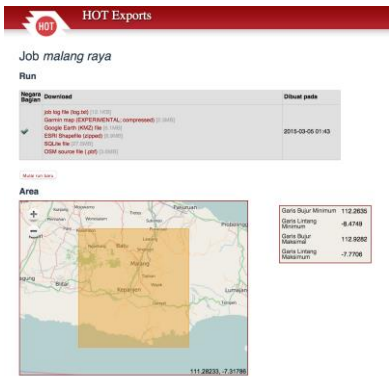
3 Mendapatkan Data

Pertama, mengeksport adalah mengkonversi data OpenStreetMap dari format XML aslinya ke data yang Anda inginkan. Ini sedikit berbeda dengan mengekstrak data, yaitu memotong data dari daerah yang Anda pilih. Ini juga berarti mengeluarkan fitur-fitur tertentu yang Anda inginkan dari sebuah daerah. Kita akan serng menggunakan istilah-istilah ini pada

bab ini, sehingga hal tersebut penting untuk dipahami perbedaannya.

Humanitarian OpenStreetMap Team (HOT)

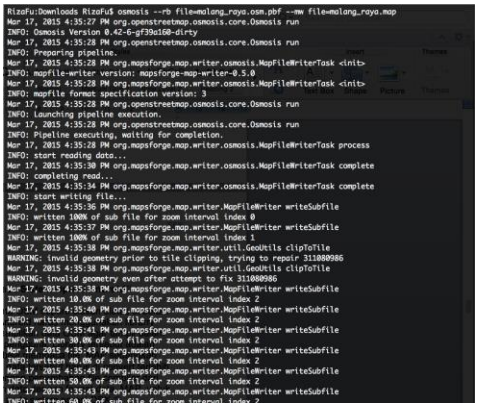
Humanitarian openstreetmap team (HOT) adalah salah satu aplikasi berbasis web yang memperbolehkan penggunanya untuk memperoleh data OpenStreetMap untuk berbagai wilayah. Pengguna dapat mengatur wilayah yang diinginkan beserta sebuah daftar fitur (tag OpenStreetMap) dalam prosesnya. Satu data yang diperoleh untuk wilayah tersebut akan tersedia dalam berbagai format untuk anda dalam beberapa menit. Dibawah ini adalah salah satu cara untuk mendapatkan data OSM pada area malang dan data yang tertera pada halaman tersebut sudah di perbarui pada tanggal 05-03-2015.



Gambar Proses pengambilan data pada HOT Exports

Menerapkan Data OSM Pada Sistem Oprasi Android

Setelah mendapatkan data OSM seperti ulasan diatas sebelum menerapkan pada sistem oprasi Android harus menggunakan librarry khusus untuk membaca dan menampilkan data OSM agar tampil dalam bentuk peta yang bisa dilihat oleh pengguna. Librarry tersebut adalah Mapsforge dan Mapsforge hanya bisa membaca file .map sedangkan data dari HotExport berupa .osm, data tersebut harus di convert terlebih dahulu menggunakan Osmosis dengan penambahan file compiler dari Mapsforge sehingga mendapatkan data dengan format data .map, untuk prosesnya menggunakan terminal pada sistem oprasi OSX.



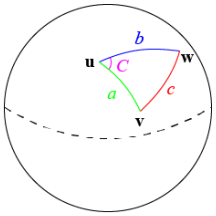
Gambar Proses konversi data peta osm ke map

Pada aplikasi Android file data .map harus diletakkan pada penyimpanan lokal di perangkat android agar bisa dibaca dan di tampilkan pada aplikasi Android tersebut. Kemudian menggunakan komponen dari library Mapsforge tersbut yang sudah di buat pada class MapView, karena Mapsforge ini OpenSource maka projek ini disertai dengan contoh aplikasinya dan siapa saja bisa mengambil dan mengembangkan contoh aplikasi tersebut, dibawah ini adalah hasil implementasi library Mapsforge yang menampilkan data peta dari OpenStreetMap yang sudah di konversi ke dalam bentuk file yang bisa ditampilkan pada library Mapsforge.

Gambar Hasil implementasi peta OSM pada Android

Haversine Formula

Teorema Haversine Formula adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dari longitude dan latitude. Ini merupakan bentuk persamaan khusus dari trigonometri bola, law of haversines, mencari hubungan sisi dan sudut pada segitiga dalam bidang bola.



Gambar Ilustrasi Spherical law of cosines

Dari gambar diatas dapat dihasilkan persamaan *Spherical law of cosines* sebagai berikut.

$$\cos(c) = \cos(a)\cos(b) + \sin(a)\sin(b)\cos(C).$$

Gambar pherical law of cosines

Dimana a,b,c ialah jarak yang bersatuan radian/sudut karena berada dalam bidang bola, yang bisa kita korelasikan dengan persamaan busur seperti gambar dibawah ini.

$$\theta = \frac{L}{r}$$

Gambar Rumus BusurDari

persamaan-persamaan di atas bisa di implementasikan pada persamaan Haversine sebagai berikut.

$$\text{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2}$$

Gambar Rumus Haversin

Untuk mendapatkan jarak di antara dua titik koordinat pada bumi, maka harus mengimplementasikan dari persamaan haversin dan *Spherical law of cosines* sehingga mendapatkan haversine formula sebagai berikut.

$$jarak = 2r \cdot \arcsin\left(\sqrt{\sin\left(\frac{Lat2 - Lat1}{2}\right)^2 + \cos(Lat2) \cdot \cos(Lat1) \cdot \sin\left(\frac{Lon2 - lon1}{2}\right)^2}\right)$$

Gambar Haversine formula

Dalam penerapannya terhadap bumi, haversine formula ini harus dikalikan dengan jari-jari dari lingkaran bumi yang nilainya 6371 km. Untuk nilai latitude dan longitude yang berbentuk derajat desimal maka harus di udah menjadi radians dengan cara mengkalikan nilai latitude dan longitude dengan 1 derajat atau 0.01745329251994 rad.

$$r(bumi) = 6371km$$

$$d = 1° = 0.01745329251994rad$$

$$jarak = 2r \cdot \arcsin\left(\sqrt{\sin\left(\frac{(Lat2 - Lat1) \cdot d}{2}\right)^2 + \cos(Lat2 \cdot d) \cdot \cos(Lat1 \cdot d) \cdot \sin\left(\frac{(Lon2 - lon1) \cdot d}{2}\right)^2}\right)$$

Gambar Haversine formula terhadap bumi

```
= 2*6371*ASIN(SQRT(
SIN((0.017453293*(Lat2-Lat1))/2)^2 +
COS(Lat2*0.017453293) *
COS(Lat1*0.017453293) *
SIN((0.017453293*(Lon2-Lon1))/2)^2))
```

Gambar Haversine formula terhadap bumi dalam Excel

PEMBAHASAN

Analisa Masalah

Pada proses ini café dan lokasi awal pengguna akan di beri tanda pada peta berupa titik-titk, dan pada peta tersebut aka ada informasi mengenai jalan-jalan yang bisa di lalui untuk mendapatkan jalur terpendek, pada persimpangan jalan-jalan tersbut akan di beri titik-titik untuk menentukan simpangan mana yang akan dipilih.

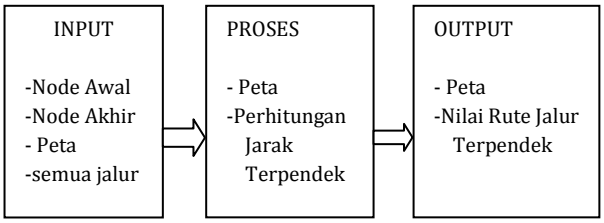
Permasalahan yang ada sebelumnya adalah rute yang menuju café tersebut bukan jalur yang terpendek, kemudian tidak ada informasi lebih lanjut terhadap café di Kota Malang. Pada Kota Malang sendiri adalah Kota yang mayoritas penduduknya adalah pendatang, sehingga mereka kurang mengetahui informasi café yang ada di Kota Malang. Adapun informasi yang ada pada website PemKot Malang.

Tetapi dalam informasi yang di sediakan tidak di sertakan peta dan jalur pertimbangan untuk

menuju café tersebut. Pada teknologi GPS saat ini sudah ada fasilitas untuk mengetahui lokasi keberadaan pengguna, tatapi untuk lokasi café tidak disertakan formasi lengkap terhadap café tersebut.

Harapannya dari analisa ini adalah menghasilkan data peta beserta titik-titik yang sudah di buat dan rute jalur terpedek menuju café dari lokasi pengguna. Sehinga jalur terpendek menuju café tersebut bisa bermanfaat untuk pengguna agar lebih mudah menuju café dan mengetahui informasi tentang café tersebut.

Analisis Sistem



Gambar Gambar Blok Diagram Sistem

Analisa Kebutuhan

Analisa kebutuhan bertujuan untuk memperoleh informasi mengenai kebutuhan input yang dibutuhkan untuk proses penentuan jalur terpendek (*shorted path*) sehingga menjadi data yang berharga untuk bisa mendapatkan hasil yang maksimal dari sekian jalur yang ada yang berada pada peta data. Kebutuhan proses yaitu kebutuhan akan proses yang terjadi sebelum dilakukannya proses algoritma, dan kebutuhan output yang dibutuhkan dari sistem aplikasi yang ada.

1. Kebutuhan Input

Kebutuhan masukan atau input yang dimaksudkan disini yaitu kebutuhan input apa saja yang dibutuhkan pada saat proses pengerjaan Algoritma *Bellman-Ford* dan sistem. Berikut ini adalah yang dibutuhkan:

a. Kebutuhan input dari amin : Hal ini bertujuan untuk merekap data-data yang terdapat di luar system untuk di gunakan sebagai bahan pembuatan system agar berjalan dengan baik. Data-data tersebut adalah data café di Kota Malang yang di masukkan pada basis data, peta Kota Malang yang di ambil dari *open street map*.

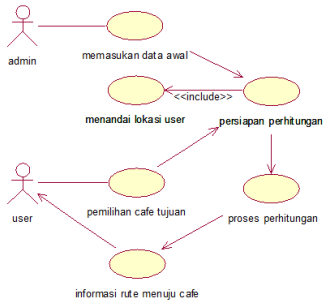
b. Kebutuhan input dari usir Untuk user sendiri tidak memerlukan inputan yang banyak, yaitu hanya memilih café yang di inginkan, atau jika user tidak memilih café tersebut maka sistem yang akan memilihkan café yang dituju.

c. Kebutuhan input dari sistem : Sistem akan memberikan inputan data sebagai persiapan perhitungan yang akan di kerjakan, yaitu berupa pencarian lokasi user yang di dapat dari GPS yang terdapat pada *smartphone* user dan penandaan berupa titik, penentuan lokasi café yang dituju serta penandaan berupa titik, pemberian titik-titik pada peta terhadap persimpangan jalan yang akan dilalui, menentukan jarak jalan yang terdapat pada peta, semua koordinat yang di beri titik pada peta di dapat dari GPS.

2. Kebutuhan Output

Hasil dari proses ini adalah berupa peta yang terdapat rute-rute yang akan dilalui serta jarak dari tiap rute-rute tersebut, serta informasi café yang akan dituju berupa infromasi text.

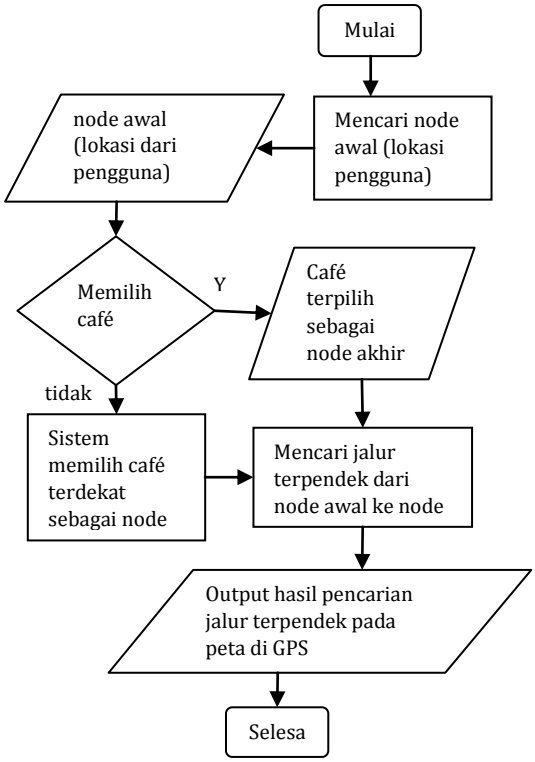
Use Case



Gambar Use case

Perancangan Sistem

1. Flowchart Sistem



Gambar Flowchart Sistem

Berikut ini adalah penjelasan secara Algoritma dari flowchart penentuan jalur terpendek :

1. Mulai.
2. Pencarian lokasi pengguna pada Peta.
3. Membaca data seluruh node dan bobot edge pada database node.
4. node awal (lokasi pengguna) dan node tujuan (café tujuan).
5. Memeriksa jalur *shorted path* apakah *query* jalur dengan node awal dan node tujuan telah tersimpan di database.
6. Melakukan proses algoritma *Bellman-Ford* dengan input node.
7. Melakukan *query* data pada peta.
8. Menampilkan hasil perhitungan dari pencarian jalur terpendek.
9. Menampilkan jalur terpendek pada peta.
10. Selesai .

STRUKTUR BASIS DATA

Berikut ini adalah penjelasan mengenai struktur basis data dalam perancangan sistem penentuan jarak terpendek menuju café di kota Malang. Tabel- tabel yang akan digunakan dalam sistem ini adalah :

1. Tabel café

Merupakan tabel yang akan digunakan untuk menyimpan data- data informasi café yang ada di kota Malang. Berikut ini adalah struktur tabelnya :

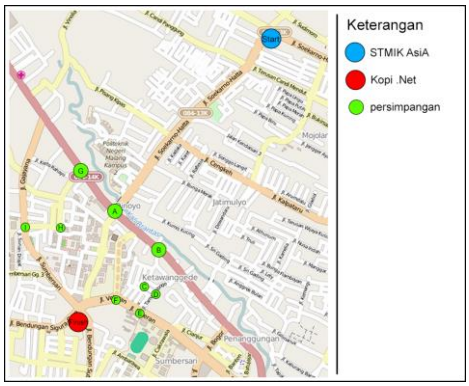
Tabel Cafe

NO	Field Name	Type	Size	Ket
1.	Id_Cafe(Primary Key)	Integer	10	
2.	Nama_cafe	Char	35	
3.	Alamat	Char	70	
4.	Jam_buka	Time		
5.	Jam_tutup	Time		
6.	Fasilitas	Char	10	
7.	Harga_menu_Min	int		
8.	Harga_menu_max	Int		
9.	Latitude	Double		Lokasi gps
10	Longitude	Double		Lokasi gps

SIMULASI PERHITUNGAN ALGORITMA

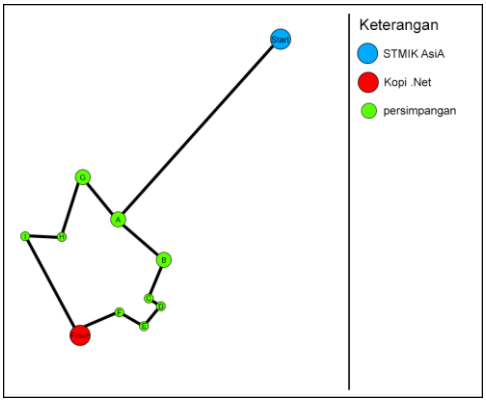
Algoritma *Bellman-Ford* menghitung jarak terpendek (dari satu sumber) pada suatu digraf berbobot. Maksudnya dari satu sumber ialah bahwa ia menghitung semua jarak terpendek yang berawal dari satu titik node.

Gambar berikut ini adalah peta yang di ambil dari open street maps sebagai contoh simulasi untuk melakukan suatu perhitungan jarak terpendek (*Shortest path)* dari node awal ke node akhir. Untuk node awal di asumsikan berada di STMIK Asia, dan untuk node akhir berada di café kopi.net. Dalam perhitungan ini, diambil beberapa titik persimpangan untuk mempermudah perhitungan secara manual agar lebih mudah dipahami.



Gambar Peta simulasi

Selanjutanya menentukan jalur-jalur yang mungkin terpendek dimana jalan dan persimpangan tersebut memungkinkan untuk di lewati. Pada kambar graf dibawan ini, belum dihitung jarak antara node/titik yang ada sesuai jalur yang mengubungkannya.



Gambar Graf perhitungan awal

Dalam gambar graf di atas sudah diketahui nilai posisi pada masing-masing node dengan format degree (DDD). Nilai posisi tersebut bisa didapat pada situs open street map, nilai posisi tersebut terdata pada tabel dibawah ini.

Tabel nilai koordinat

Node	Latitude	Longitude
Start	-7.937862	112.626014
A	-7.949908	112.615494
B	-7.952624	112.618664
C	-7.955259	112.617763
D	-7.955572	112.618192
E	-7.957001	112.617173
F	-7.955997	112.615338
G	-7.946885	112.612907
H	-7.950971	112.611351
I	-7.950899	112.608942
Finish	-7.956866	112.612817

Berikutnya menentukan bobot masing-masing (*edge*), agar dalam perhitungan jarak menuju node akhir bisa menghasilkan nilai yang akurat dan bisa di bandingkan dengan perihtungan yang lain. Dimana perhitungan *edge* tersebut menggunakan *Haversine formula* terhadap setiap titik yang terhubung oleh garis lurus. Perhitungan yang pertama dimulai dari node Start ke node A mengunakan *Haversine Formula*. Koordinat Start sebagai Latitude 1 dan Longitude 1, sedangkan koordinat A sebagai Latitude 2 dan Longitude 2. Nilai radian dan jari-jari bumi sudah diketahui sebelumnya, untuk penjelasan perhitugan sebagai berikut.

r(bumi) = 6371 km
d = 0.01745329251994 rad
Koordinat Start
Lat1 = -7.937862
Long1 = 112.626014
Koordinat A
Lat2 = -7.949908
Long2 = 112.615494
ΔLat = (Lat2 – Lat1) x d
= (-7.949908 - -7.937862) x 0.01745329251994 = -2.10E-04
ΔLong = (Long2 – Long2) x d
= (112.615494 - 112.626014) x 0.01745329251994 = -0.000183609
$\sin(\Delta Lat/2)^2 = \sin(-2.10E-04 / 2)^2 =$ 1.11E-08
$\sin(\Delta Long/2)^2 = \sin(-0.000183609 / 2)^2 =$ 8.42807E-09

cos(Lat2 x d) = cos(-7.949908 x 0.01745329251994) = 0.990389365
cos(Lat1 x d) = cos(-7.937862 x 0.01745329251994) = 0.990418421

Jarak = 2r x arcsin(√sin(ΔLat/2)² + cos(Lat2 x d) x cos(Lat1 x d) x sin(ΔLong/2)²)
Jarak = 2 x 6371 x arcsin(√1.11E-08 + 0.990389365 x 0.990418421 x 8.42807E-09)
Jarak = 1,770978449 km = 1,771 km

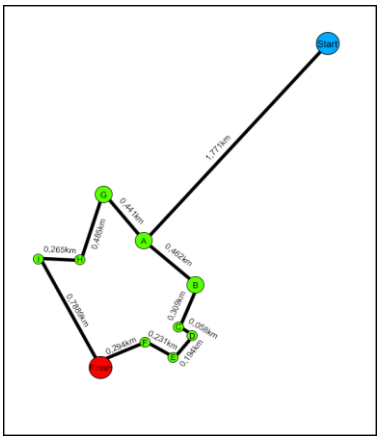
Dari perhitungan di atas, dihasilkan nilai jarak antara koordinat Start dan koordinat A yaitu sejauh 1,771 km dan dibawah ini adalah tabel hasil perhitungan pertama.

Tabel Perhitungan jarak Start ke A

Start ke A	Start	A	Jarak (kilo meter)	Jarak (meter)
Latitude	-7.937862	-7.949908	1.770978449	1770.978449
Longitude	112.626014	112.615494		

Untuk perhitungan selanjutnya agar mempermudah dan mempercepat perhitungan bisa menggunakan bantuan Ms.Excel dan Haversine formula terhadap bumi untuk Ms.Excel.

Dari hasil seluruh perhitungan diatas, maka gambar graf yang sebelumnya tidak ada jarak antara node sekarang sudah ada jarak yg menghubungkan antara node tersebut. Sehingga bisa dilakukan perhitungan pencarian jalur terpendek menggunakan algoritma Bellman-Ford.



Gambar Graf jalur yang akan di lalui

Untuk memulai perhitungan menggunakan algoritma Bellman-Ford node awal harus diberikan nilai 0 dan node yang lain hingga node akhir diberikan nilai ∞.

Tabel Awal perhitungan algortima Bellman-Ford

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0										
2	0										
3	0										
4	0										
5	0										
6	0										
7	0										

Kemudian dalam perhitungan ini menggunakan rumus Bellman-Ford yang sesuai pada landasan teori yaitu

$$M[i,v] = \min(M[i-1,v], (M[i-1,n] + C_{vn}))$$

i = iterasi, v = vertex = node, n = node neighbor, C = cost

Saat ini perhitungannya untuk menentukan iterasi yang pertama dimana nilai dari masing-masing node masih ∞ dan akan di tambahkan *cost* atau bobot jarak yang ditempuh dari node awal ke node tujuan. Berikut perhitugan untuk iterasi yang pertama.

$M[1,A] = \min(M[0,A],(M[0,Start)+C_{startA})$
$= \min(\infty,(0+1{,}771km)) = \min(\infty,1{,}771km)$
$= 1{,}771km$

Tabel Perhitungan algortima Bellman-Ford iterasi 1

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0										
3	0										
4	0										
5	0										
6	0										
7	0										

Dalam perhitungan iterasi pertama diketahui nilai bobot node A sebesar 1,771km sedangkan node yang lain masih bernilai ∞, karena dalam perbandingan perhitunga sesuai rumus jika di hitung akan tetap bernilai ∞ dikarenakan yang terhubung langsung terhadap node Start hanya node A. Selanjutnya perhitungan iterasi 2.

M[2,B] = min(M[1,B] , (M[1,A]+C_{AB}))
= min(∞ , (1,771km + 0,462km))
= min(∞ , 2,233km) = 2,233km
M[2,G] = min(M[1,G] , (M[1,A]+C_{AG}))
= min(∞, (1,771km + 0,441km))
= min(∞, 2,212km) = 2,212km

Tabel Perhitungan algortima Bellman-Ford iterasi 2

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0										
4	0										
5	0										
6	0										
7	0										

Perhitungan iterasi kedua ini bisa di lihat bahwa nilai node B dan F bisa di ketahui sedangkan yang lain masih tetap ∞ karena nilai node tetangganya terdekat masih bernilai ∞. Selanjutanya perhitungan iterasi ketiga.

M[3,C] = min(M[2,C], (M[2,B]+C_{CB}))
= min(∞,(2,233km + 0,309km))
= min(∞,2542km) = 2,542km
M[3,H] = min(M[2,H], (M[2,G]+ C_{HG}))
= min(∞,(2,212km + 0,485km))
= min(∞, 2,697km) = 2,697km

Tabel Perhitungan algortima Bellman-Ford iterasi 3

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0	1,771	2,233	2,542	∞	∞	∞	2,212	2,697	∞	∞
4	0										
5	0										
6	0										
7	0										

Perhitungan iterasi ke tiga sudah mulai terlihat jelas nilai-nilai bobot pada node-node yang ada. Berikutnya iterasi ke empat

M[4,D] = min(M[3,D], (M[3,C]+C_{DC}))
= min(∞,(2,542km + 0,058km))
= min(∞, 2,6km) = 2,6km
M[4,I] = min(M[3,I], (M[3,H]+ C_{IH}))
= min(∞, (2,697km + 0,265km))
= min(∞, 2,962km) = 2,962km

Tabel perhitungan algortima Bellman-Ford iterasi 4

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0	1,771	2,233	2,542	∞	∞	∞	2,212	2,697	∞	∞
4	0	1,771	2,233	2,542	2,6	∞	∞	2,212	2,697	2,962	∞
5	0										
6	0										
7	0										

Perhitungan iterasi ke empat menunjukan nilai node E, F dan Finish tetap ∞ karena nilai pembanding di iterasi ketiga masih ∞. Berikutnya perhitungan iterasi kelima.

M[5,E] = min(M[4,E] , (M[4,D] + C_{ED}))
= min(∞, (2,6km + 0,194km))
= min(∞, 2,794km) = 2,794km
M[5,Finish] = min(M[4,Finish] , (M[4,I] + $C_{FinishI}$))
= min(∞, (2,962km + 0,789km))
= min(∞, 3,751km) = 3,751km

Tabel perhitungan algortima Bellman-Ford iterasi 5

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0	1,771	2,233	2,542	∞	∞	∞	2,212	2,697	∞	∞
4	0	1,771	2,233	2,542	2,6	∞	∞	2,212	2,697	2,962	∞
5	0	1,771	2,233	2,542	2,6	2,794	∞	2,212	2,697	2,962	3,751
6	0										
7	0										

Pada iterasi kelima sudah mendapatkan nilai dari node finish atau node tujuan. Akan tetapi semua node belum terhitung seluruhnya. Berikutnya perhitungan untuk iterasi yang ke enam dan tujuh.

$$M[6,F] = \min(M[5,F], (M[5,E] + C_{EF}))$$
$$= \min(\infty, (2{,}794\text{km} + 0{,}231\text{km}))$$
$$= \min(\infty, 3{,}025\text{km}) = 3{,}025\text{km}$$

Tabel perhitungan algortima Bellman-Ford iterasi 6

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0	1,771	2,233	2,542	∞	∞	∞	2,212	2,697	∞	∞
4	0	1,771	2,233	2,542	2,6	∞	∞	2,212	2,697	2,962	∞
5	0	1,771	2,233	2,542	2,6	2,794	∞	2,212	2,697	2,962	3,751
6	0	1,771	2,233	2,542	2,6	2,794	3,025	2,212	2,697	2,962	3,751
7	0										

$$M[7,Finish] = \min(M[6,Finish], (M[6,F] + C_{FinishF}))$$
$$= \min(3{,}751\text{km}, (3{,}025\text{km} + 0{,}294\text{km}))$$
$$= \min(3{,}751\text{km}, 3{,}319\text{km}) = 3{,}319\text{km}$$

Tabel perhitungan algortima Bellman-Ford iterasi 7

Node (km)

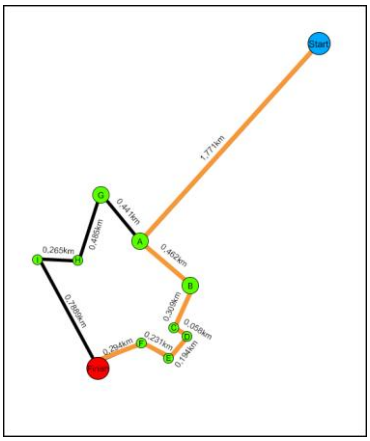
Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0	1,771	2,233	2,542	∞	∞	∞	2,212	2,697	∞	∞
4	0	1,771	2,233	2,542	2,6	∞	∞	2,212	2,697	2,962	∞
5	0	1,771	2,233	2,542	2,6	2,794	∞	2,212	2,697	2,962	3,751
6	0	1,771	2,233	2,542	2,6	2,794	3,025	2,212	2,697	2,962	3,751
7	0	1,771	2,233	2,542	2,6	2,794	3,025	2,212	2,697	2,962	3,319

Dari hasil iterasi ketujuh sudah didapatkan nilai akhir dari seluruh perhitungan sehingga sudah ditentukan jalur mana yang akan dilalui. Berikut hasil akhir perhitungan.

Tabel hasil perhitungan algortima Bellman-Ford

Node (km)

Iterasi	Start	A	B	C	D	E	F	G	H	I	Finish
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1,771	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	1,771	2,233	∞	∞	∞	∞	2,212	∞	∞	∞
3	0	1,771	2,233	2,542	∞	∞	∞	2,212	2,697	∞	∞
4	0	1,771	2,233	2,542	2,6	∞	∞	2,212	2,697	2,962	∞
5	0	1,771	2,233	2,542	2,6	2,794	∞	2,212	2,697	2,962	3,751
6	0	1,771	2,233	2,542	2,6	2,794	3,025	2,212	2,697	2,962	3,751
7	0	1,771	2,233	2,542	2,6	2,794	3,025	2,212	2,697	2,962	3,319

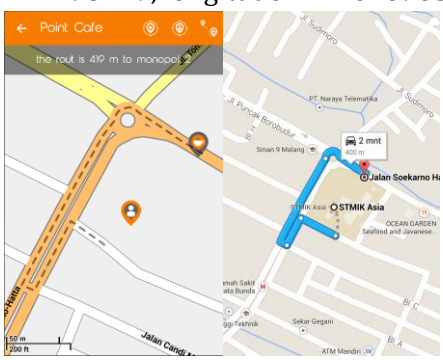


Gambar Graf jalur yang dari hasil perhitungan

Pengujian Akurasi Rute

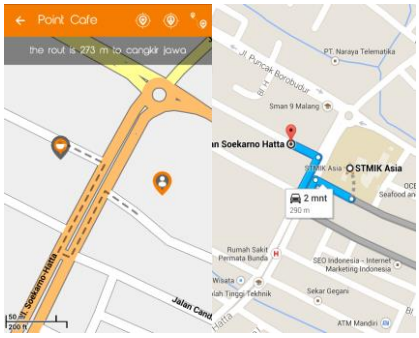
Pengujian ini dilakukan berdasarkan posisi user degan café menggunakan perhitungan berdasarkan metode *Bellman-Ford* dengan hasil yang dibandingkan dengan google maps. Posisi user berada di STMIK ASIA dengan Latitude -7.937763 dan Longitude 112.626370.

1. Pengujian pertama, latitude: -7.93729, longitude: 112.626965.



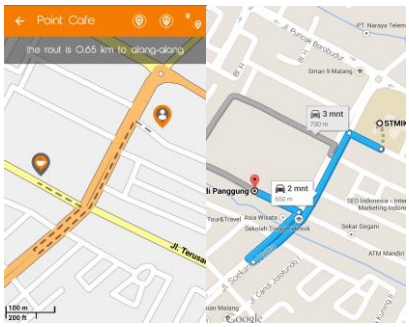
Gambar Perbandingan akurasi pertama

2. Pengujian kedua, latitude: -7.937563, longitude: 112.625428.



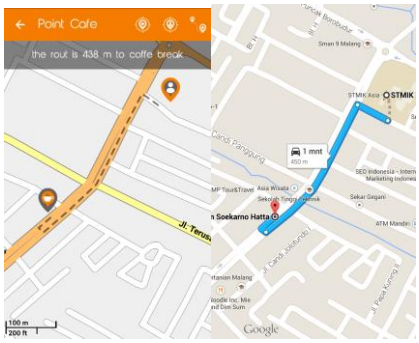
Gambar Perbandingan akurasi kedua

3. Pengujian ketiga, latitude: -7.938925, longitude: 112.624055.



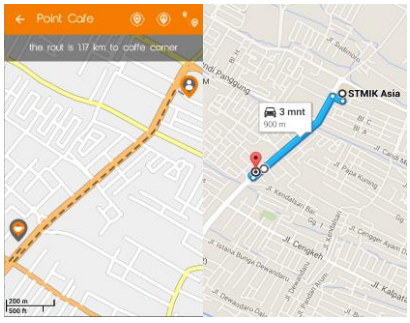
Gambar Perbandingan akurasi ketiga

4. Pengujian ke empat, latitude: -7.94003, longitude: 112.624092.



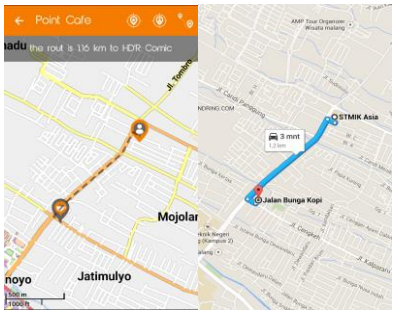
Gambar Perbandingan akurasi ke empat

5. Pengujian kelima, latitude: -7.943793, longitude: 112.619744.



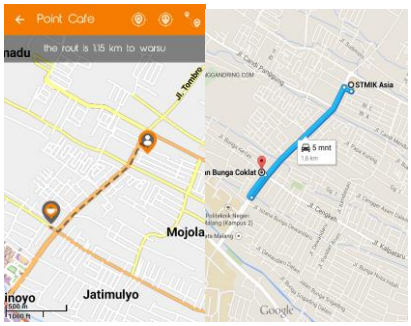
Gambar Perbandingan akurasi kelima

6. Pengujian ke enam, latitude: -7.945033, longitude: 112.620087.



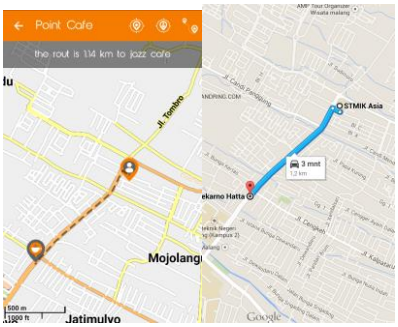
Gambar Perbandingan akurasi ke enam

7. Pengujian ketujuh, latitude: -7.94409, longitude: 112.618981.



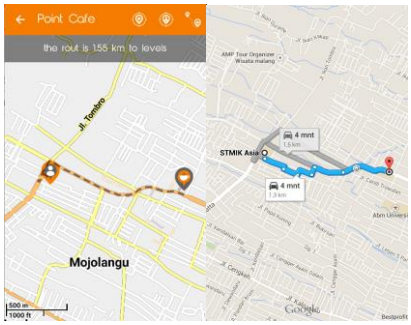
Gambar Perbandingan akurasi ketujuh

8. Pengujian kedelapan, latitude: -7.945041, longitude: 112.619161.



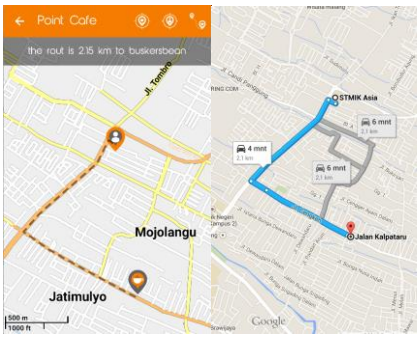
Gambar Perbandingan akurasi kedelapan

9. Pengujian kesembilan, latitude: -7.939259, longitude: 112.636582.



Gambar Perbandingan akurasi kesembilan

10. Pengujian kesepuluh, latitude: -7.949241, longitude: 112.628008.



Gambar Perbandingan akurasi kesepuluh

Dari pengujian diatas tersebut berikut adalah table dan analisanya :

Tabel 4.2 Hasil pengujian keakuratan

No	Koordinat café		Nama Café	Jarak	Akurat
	Latitude	Longitude			
1	-7.93729.	112.626965	Monopoli 2	419m	Y
2	-7.937563	112.625428.	Cangkir jawa	279m	Y
3	-7.938925	112.624055	Alang-alang	0,65km	Y
4	-7.94003	112.624092	Coffe break	438m	Y
5	-7.943793	112.619744	Coffe corner	1,17km	Y
6	-7.945033	112.620087	HD'R comic	1,16km	Y
7	-7.94409	112.618981	Warsu	1,15km	Y
8	-7.945041	112.619161	Jazz café	1,14km	Y
9	-7.939259	112.636582	Levels	1,55km	T
10	-7.949241	112.628008	Buskerbean	2,15km	Y

Dari pengujian di atas maka dapat disimpulkan system berjalan baik dengan 9 nilai akurat, prosentasenya adalah 9/10 x100 = 90%. Factor ketidak akuratan adalah tidak bisa memberikan rute melalui jalan alternative

PENUTUP

KESIMPULAN

Berdasaarkan penjelasan serta uraian yang telah dibahas oleh penulis pada laporan tugas akhir ini, maka dapat diperoleh beberapa kesimpulan, diantaranya :

1. Dengan dibangunya aplikasi *Point Cafe*, dapat membantu pengguna dalam memilih café yang terdapat di kota malang dengan lebih mudah dan aplikatif.
2. Dalam pencarian café di kota malang ini sebagai masukan yang berupa titik koordinat user sebaiknya dilakukan di luar ruangan atau *outdoor* untuk tingkat akurasi penitikan koordinat user yang baik.
3. Tingkat kecepatan untuk menampilkan *output* berupa Maps ini bisa di bilang cepat, karena berjalan dalam kondisi *off line*.
4. Untuk penambahan cafe pada aplikasi ini masih belum bisa, karena membutuhkan biaya dan waktu yang cukup untuk membangun sistem yang terintegrasi secara.client server..

SARAN

Mengingat berbagai keterbatasan dari penulis, maka berikut saran yang nantinya bisa membantu atau membangun suatu sistem yang lebih baik lagi :

1. Diharapkan utuk dapat menambah variabel atau keriteria agar dapat menghasilkan informasi yang lebih sesuai.
2. Dengan adanya sistem yang telah dibangun hanya single user hendaknya bagi penulis selanjutnya dapat mengembangkan sistem baru yang menggunakan multi user agar user bisa menambahkan cafe user sendiri sesuai kemauan user.

DAFTAR PUSTAKA

[1] Aditya Pradhana, Bayu. Studi dan Implementasi Persoalan Lintasan Terpendek Suatu Graf dengan Algoritma Djikstra dan Algoritma Bellman-Ford. http://repository.usu.ac.id. 2006

[2] Chamero, Juan. Dijkstra's Algorithm. http://www.intag.org/downloads/ ds_006.pdf. 2006

[3] Dwi Anggara, Fajar. Studi dan Implementasi Struktur Data Graf. http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2008-2009/Makalah2008/Makalah0809-097.pdf. 2008

[4] Elcom, Exploring Android On Your Own PC, Jakarta : Andi, 2012

[5] Hougardy, Stefan. The Floyd-Warshall Algorithm on Graphs with Negative Cycles. http://www.or.uni-bonn.de/~hougardy/paper/Floyd-Warshall.pdf. 2010

[6] Khairurrazi Budiarsyah, Dibi. Algoritma Djikstra, Bellman-Ford, Dan Floyd-Warshall Untuk Mencari Rute Terpendek Dari Suatu Graf. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. 2010

[7] Komputer Wahana, Membuat Aplikasi Android untuk Tablet dan Handphone , Jakarta : Elex Media Komputindo, 2012

[8] Priyanta, Pemrograman Android Untuk Pemula, Jakarta : Cerdas Pustaka Publisher, 2011

[9] Purwananto, Yudhi. Implementasi Dan Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya. Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember. 2005

[10] Ramadhan, Fahmi. Algoritma Bellman-Ford dan Floyd-Warshall. http://fahmiramadhan.files.wordpress.com/2009/03/algoritma-floyd-warshall-dan-bellman-ford.pdf. 2009