

Pencarian Rute Perjalanan Terpendek pada Open Street Map dengan Algoritma A*

Benardi Atmadja¹, Rinaldi Munir²

Informatics/Computer Science
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia

13510078@std.stei.itb.ac.id¹, rinaldi@informatika.org²

Abstrak— Open Street Map (OSM) merupakan layanan peta luring yang sedang berkembang. Open Street Map memiliki berbagai potensi yang dapat dikembangkan, salah satunya adalah fitur pencarian rute terpendek. Pencarian rute terpendek berarti mencari rute dari suatu titik awal menuju titik akhir yang menghasilkan bobot pencarian terkecil.

Pada Tugas Akhir ini dilakukan studi untuk mencari rute terpendek dari data OSM secara daring. Data OSM berbentuk file XML yang berisi titik-titik pada peta dan keterhubungan antar titik tersebut. Data tersebut diunduh dan diparsing untuk membentuk graf berarah. Setiap sisi pada graf berarah tersebut dihitung bobotnya dengan formula Haversine. Formula Haversine adalah perhitungan jarak terpendek dari dua buah titik yang memiliki bujur dan lintang pada bidang yang berdimensi bola. Graf yang telah memiliki bobot digunakan untuk penelusuran pencarian rute terpendek. Program menerima input berupa simpul awal dan simpul tujuan. Algoritma A* digunakan untuk menelusuri graf yang sudah terbentuk. Hasil dari penelusuran algoritma A* berupa file teks berisi rute terpendek yang dapat ditampilkan pada perangkat visualisasi.

Kasus uji yang digunakan untuk memeriksa kebenaran pencarian rute perjalanan terpendek adalah kasus dengan jalan satu arah, kasus dengan jalanan yang jauh, dan kasus jalanan yang terpotong akibat pembatasan wilayah pencarian. Pengecekan validitas pencarian rute terpendek dilakukan dengan membandingkan hasil rute program dengan hasil rute Google Maps.

Hasil yang didapatkan menunjukkan bahwa program sudah dapat melewati kasus uji dengan baik. Pada kasus uji satu arah, program menghasilkan rute perjalanan yang tidak melanggar jalur pada peta. Pada jalur jalanan yang jauh, program berhasil menghasilkan rute perjalanan terpendek yang sesuai dengan Google Maps.

Kata kunci — Algoritma A*, Google Maps, OpenStreetMap, Pencarian rute terpendek, XML

I. PENDAHULUAN

Pada umumnya manusia menggunakan peta untuk mencari rute perjalanan dari suatu tempat ke tempat lainnya. Penggunaan peta sendiri sudah berkembang dari mulanya yang berupa fisik

hingga sekarang sudah berupa layanan peta yang dapat diakses secara online.

Situs layanan penyedia yang terkenal dan biasa diakses salah satunya adalah Google Maps. Google Maps telah mencakup peta sebagian besar wilayah di dunia termasuk di Indonesia. Kekurangan dari Google Maps adalah tidak lengkapnya data-data wilayah yang masih terpendel.

Situs layanan penyedia peta yang saat ini sedang berkembang adalah Open Street Map (OSM). Situs OSM sudah memiliki database untuk wilayah Indonesia dan dapat diakses melalui www.openstreetmap.org. Saat ini OSM telah digunakan oleh banyak sekali pengguna peta secara online di dunia untuk kebutuhan tertentu. Karakteristiknya yang terbuka dan gratis mendorong pengguna untuk memakai layanan peta tersebut. Akan tetapi fitur pencarian rute terpendek pada OSM belum tersedia.

Pada Tugas Akhir ini dibuat fitur pencarian rute terpendek dengan menggunakan OSM sebagai data utamanya. Data peta dari peta OSM digunakan sebagai database untuk membangun struktur graf berbobot dan berarah untuk pencarian rute terpendek dari suatu tempat ke tempat yang lain.

Masalah paling utama dalam pembuatan fitur pencarian rute perjalanan adalah bagaimana implementasi algoritma agar sesuai dengan fungsionalitas yang diharapkan. Selain itu perlu dilakukan penyesuaian pada database yang sudah tersedia dari OSM agar dapat melakukan pencarian rute terpendek.

II. RISET TERKAIT

Berikut merupakan review dari riset yang terkait:

A. Algoritma A* untuk Penentuan Rute Terpendek Sepeda

Pepping (2009) membahas mengenai metode penentuan rute terpendek untuk sepeda di daerah Netherlands. Permasalahan yang dihadapi adalah belum adanya suatu layanan peta yang menyediakan pencarian rute perjalanan terpendek untuk sepeda. Algoritma yang digunakan adalah algoritma A* yang dan Dijkstra sebagai perbandingan. Hasil yang didapatkan adalah sebagai berikut

Tabel 1. Hasil perbandingan algoritma A* dengan Dijkstra

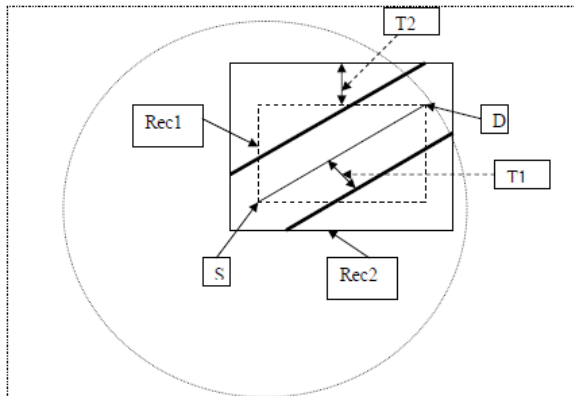
(Sumber: Pepping 2009. Halaman 33)

From	To	Distance (km)	A* (s)	Dijkstra (s)	Speed up
Amsterdam	Utrecht	44.18	0.13	1.36	10.5
Alkmaar	Breda	155.49	2.53	3.71	1.5
Leeuwarden	Rotterdam	211.39	2.15	5.28	2.5
Den Helder	Tilburg	215.18	3.24	7.05	2.2
Groningen	Tilburg	270.53	4.19	7.59	1.8
Alkmaar	Maastricht	301.67	6.85	9.23	1.3
Den Helder	Maastricht	342.51	7.4	9.04	1.2
Groningen	Middelburg	364.41	7.55	8.46	1.1

Dari tabel 1 terlihat bagaimana algoritma A* yang digunakan menghasilkan rute terpendek yang lebih cepat dibandingkan Dijkstra. Namun semakin jauh jarak pencarian maka hasil yang didapat semakin mirip dengan algoritma Dijkstra.

B. Algoritma A* dengan Pembatasan Wilayah untuk Pencarian Rute

Liang (2005) membahas implementasi algoritma A* dengan pembatasan wilayah *bounding box* pada peta kota Ottawa, Kanada.



Gambar 1. *Bounding box* yang digunakan (Liang 2005)

Liang menggunakan *bounding box* berupa segi empat yang didapatkan dari titik awal lokasi pencarian ke titik tujuan. Algoritma A* yang digunakan mencari hanya dalam luas wilayah *rec2* pada gambar 1. Perhitungan jarak yang digunakan adalah Jarak Euclidean yaitu mencari jarak antara 2 buah titik dengan cara Pythagoras.

Tabel 2. Hasil perbandingan pencarian rute terpendek dengan menggunakan algoritma Dijkstra, A*, dan A* yang dibatasi (*restricted*)

(Sumber: Liang 2005. Halaman 12)

Distance	5009	7527	11262	13957	15164	19366
Dijkstra	25/20	50/45	55/50	71/70	100/80	160/140
A*	24/19	40/38	45/40	50/45	70/60	110/100
Accuracy	1.0	1.0	1.0	1.0	1.0	1.0
Restricted	12/10	15/10	20/15	20/18	40/30	60/54
Accuracy	1.28	1.26	1.02	1.25	1.26	1.1

Hanya saja Liang menyebutkan bahwa pembatasan wilayah harus disesuaikan agar jalur terpendek dapat selalu ditemukan. Apabila rute tidak ditemukan perlu dilakukan pencarian dengan batas wilayah yang lebih besar. Hal ini dapat menyebabkan waktu eksekusi menjadi lebih lama.

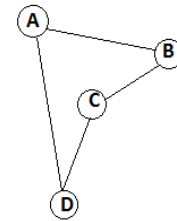
III. STUDI LITERATUR

Berikut merupakan studi literatur dalam Tugas Akhir ini:

A. Struktur data Graf

Dalam tugas akhir ini digunakan struktur data graf untuk merepresentasikan jalur lalu lintas. Berdasarkan (Rinaldi 2002) Graf disusun dari dua buah komponen utama. Simpul/ titik (*vertex* atau *V*) dan sisi (*edge* atau *E*). Suatu graf dapat disimbolkan dalam notasi $G = (V,E)$. Komponen *V* pada graf berisi kumpulan titik yang ada pada graf, dan komponen *E* berisi sisi yang menghubungkan sepasang simpul.

Graf dapat dibagi menjadi dua berdasarkan orientasi arah pada sisi, yaitu graf berarah dan graf tidak berarah. Pada graf berarah, urutan titik pada sisi berpengaruh pada graf.

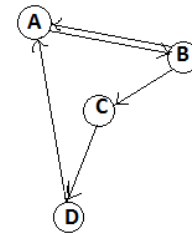


Gambar 2. Graf sederhana tak berarah

Contoh sebuah graf sederhana tak berarah (gambar II.1) dengan *V* dan *E*

$$V = \{A,B,C,D\}$$

$$E = \{(A,B),(A,D),(B,C),(C,D)\}$$



Gambar 3. Contoh graf berarah

Contoh graf berarah adalah sebagai berikut (Gambar II.2):

$$V = \{A,B,C,D\}$$

$$E = \{(A,B),(B,A),(B,C),(C,D),(D,A)\}$$

Pada contoh graf sederhana di gambar II.1 termasuk graf yang tidak berarah yang berarti sisi $(A,B) = \text{sisi } (B,A)$. Sedangkan pada contoh di gambar II.2 merupakan graf yang berarah. Di

sini sisi (A,B) dan sisi (B,A) merupakan sisi yang berbeda. Pada sisi (B,C) berarti titik B dapat mencapai C namun tidak sebaliknya karena tidak terdapat sisi (C,B).

Struktur data graf dapat digunakan untuk merepresentasikan jalan di dunia nyata dengan V mewakili kumpulan lokasi yang akan dimodelkan sedangkan E merupakan kumpulan jalan yang menghubungkan tempat-tempat tersebut. Oleh karena jalan di dunia nyata mempunyai aturan satu arah ataupun dua arah maka graf berarah yang lebih sesuai untuk digunakan.

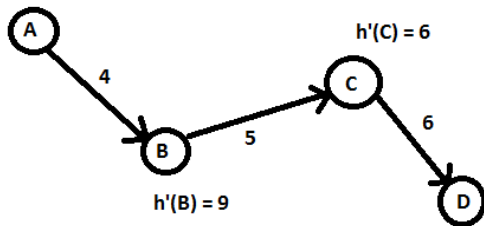
B. Algoritma A*

Berdasarkan [Liang 2005] Algoritma A* menerapkan konsep bahwa dalam suatu perjalanan ke suatu lokasi, biasanya tidak diperlukan mengambil jalan yang berlawanan arah. Algoritma A* memberikan prioritas pada rute yang mengarah ke tujuan. Algoritma ini tidak menelusuri semua wilayah pencarian.

Heuristik digunakan pada Algoritma A* untuk menjadikannya lebih berfokus ke tujuan. Hal ini mempercepat evaluasi dibandingkan mengevaluasi setiap simpul v dengan menggunakan perkiraan jalur terpendek pada simpul $g(v)$.

$$f(v) = g(v) + h(v) \quad (\text{Persamaan 1})$$

Fungsi evaluasi $f(v)$ merepresentasikan panjang dari jalur terpendek dari sumber simpul s ke simpul tujuan t ketika penelusuran dilakukan melalui simpul v . $g(v)$ merupakan jarak dari s ke v dan $h(v)$ adalah jarak dari v ke t . Estimasi pada fungsi evaluasi adalah sebagai berikut:



Gambar 4. Contoh ilustrasi algoritma A*

Pada contoh gambar 4 kita memiliki suatu graf berarah $V = \{A,B,C,D\}$ dengan $E = \{(A,B),(B,C),(C,D)\}$. Misalkan A adalah titik awal dan D adalah titik tujuan. Angka pada tiap sisi graf merupakan jarak antar dua buah titik. Nilai dari fungsi heuristik h' adalah estimasi jarak dari suatu titik pada graf ke titik tujuan D. Jadi $h'(B)$ merupakan jarak perkiraan dari B menuju D yang di sini diambil dari jarak garis lurus menuju titik D.

Perhitungan fungsi $f'(B)$ menjadi :

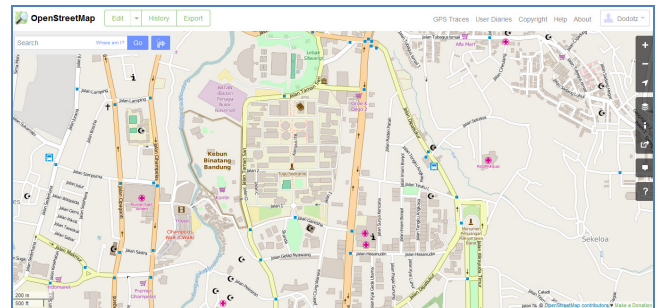
$$\begin{aligned} f'(B) &= g'(B) + h'(B) \\ &= 4 + 9 \\ &= 13 \end{aligned} \quad (\text{Persamaan 2})$$

Sedangkan fungsi $f'(C)$ adalah:

$$\begin{aligned} f'(C) &= g'(C) + h'(C) \\ &= (4+5) + 6 \\ &= 15 \end{aligned} \quad (\text{Persamaan 3})$$

C. Layanan Open Street Map

Open Street Map (OSM) merupakan suatu layanan peta online yang dibuat dan dikembangkan untuk didistribusikan secara gratis. Proyek ini dikembangkan karena sebagian besar layanan peta online yang ada mempunyai hak cipta dan batasan penggunaan berbayar sehingga menghambat orang-orang untuk menggunakannya secara kreatif maupun produktif.



Gambar 5 Tampilan layanan peta OSM

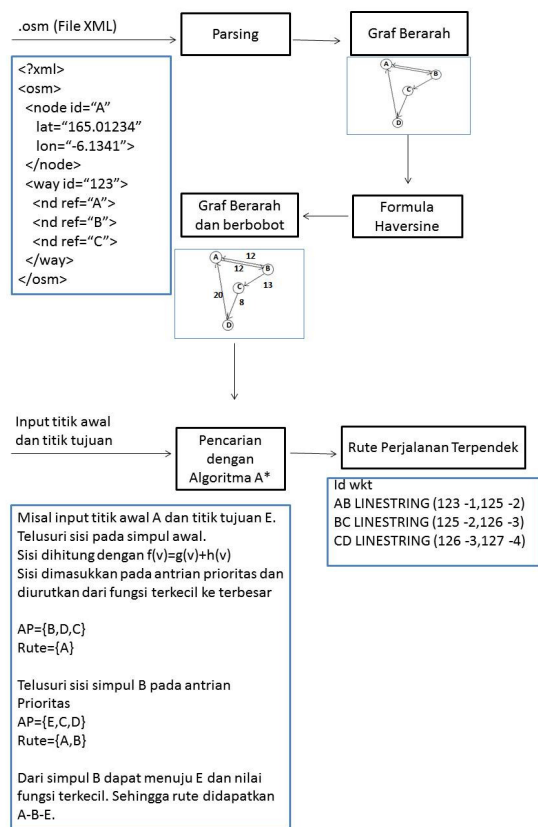
(sumber: <http://www.openstreetmap.org/#map=16/-6.8911/107.6122>)

Gambar 5 merupakan tampilan layanan peta OSM di area Bandung dengan pusat tampilan pada Institut Teknologi Bandung. Seperti yang dapat dilihat peta yang ditampilkan sudah mencakup legenda-legenda serta informasi penting yang seperti pada kondisi sebenarnya.

Kelengkapan peta merupakan hasil kontribusi dari orang yang menyunting peta. OSM menyediakan tools bagi pengguna untuk ikut serta menyunting dan melengkapi peta. Format yang digunakan pada OSM adalah file xml. File xml berisi informasi-informasi dalam bentuk teks yang diberikan <tag>. Seperti pada file osm berikut:

```
<?xml version="1.0" encoding="UTF-8" ?>
<osm version="0.6" generator="CGImap 0.4.0 (21713
thorn-02.openstreetmap.org)" copyright="OpenStreetMap and
contributors" attribution="http://www.openstreetmap.org/copyright"
license="http://opendatacommons.org/licenses/odbl/1-0/">
<bounds minlat="-6.9162000" minlon="107.6119000" maxlat="-6.9087000"
maxlon="107.6195000"/>
<node id="25433457" visible="true" version="2" changeset="839915"
timestamp="2009-03-21T14:18:04Z" user="adhitya" uid="7748" lat=
"-6.9152856" lon="107.6155852"/>
<node id="3305774350" visible="true" version="1" changeset="28348138"
timestamp="2015-01-23T12:25:50Z" user="feryseptember" uid="2366544"
lat="-6.9098523" lon="107.6158787">
<tag k="amenity" v="place of worship"/>
<tag k="name" v="Ar-Rauman"/>
</node>
<node id="3305776942" visible="true" version="1" changeset="28348138"
timestamp="2015-01-23T12:28:50Z" user="feryseptember" uid="2366544"
lat="-6.9143384" lon="107.6119380">
<tag k="amenity" v="police"/>
<tag k="name" v="office corp"/>
</node>
```

Gambar 6. Potongan isi dari file berformat osm



Gambar 7. Bagan proses solusi

Berdasarkan bagan pada gambar 7 proses pencarian rute dimulai dengan memarsing file xml berformat .osm. File xml tersebut berisi data yang menyusun peta pada Open Street Map. Pemrosesan parsing menghasilkan suatu graf berarah yang merepresentasikan jalanan dari peta OSM.

Graf berarah tersebut belum memiliki bobot karena hasil parsing masih berupa simpul dan sisi. Oleh sebab itu untuk membuat graf tersebut memiliki bobot jarak, maka digunakan formula haversine untuk menghitung jarak sisi-sisi dari setiap simpul. Keluaran dari perhitungan dengan formula haversine berupa graf berarah yang memiliki bobot.

Program menerima masukan berupa simpul awal dan simpul tujuan untuk dicari rute perjalanan terpendeknya. Graf hasil proses selanjutnya ditelusuri dengan menggunakan algoritma A*. Pada mulanya algoritma A* menelusuri sisi dari simpul awal. Sisi-sisi yang didapatkan kemudian dihitung dengan $f(v) = g(v) + h(v)$ dari persamaan 1.

Nilai dari fungsi tersebut diurutkan berdasarkan nilai terkecil ke terbesar dan sisi tersebut dimasukkan ke dalam antrian prioritas. Selanjutnya simpul yang terdapat pada antrian prioritas ditelusuri dan dicatat rutanya. Simpul tersebut ditelusuri lagi sisinya dan dihitung dengan algoritma A*. Sisi

yang sudah ditelusuri dimasukan ke dalam antrian prioritas. Proses ini terus diulang hingga titik tujuan ditemukan dan nilai fungsi dari perhitungan merupakan nilai terkecil di antrian prioritas. Apabila nilai fungsi belum merupakan yang terkecil, maka ditelusuri lagi simpul yang ada pada antrian prioritas.

Algoritma A* menghasilkan keluaran berupa file .csv yang berisi Id dan wkt. Id merupakan nama dari simpul menuju sisi yang ada pada rute perjalanan. Misalkan hasil rute perjalanan berupa A,B,D,E maka Id akan berisi AB, BD, dan DE. Sedangkan wkt berisi format yang digunakan untuk menggambar garis pada perangkat visualisasi. Formatnya adalah LINestring(LintangA BujurA, LintangB BujurB).

IV. IMPLEMENTASI DAN PENGUJIAN

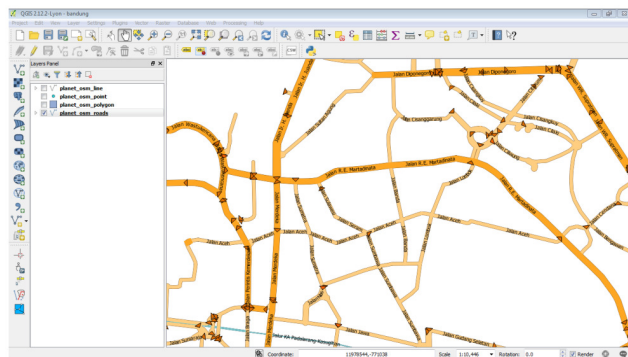
Pada bagian ini dibahas implementasi solusi serta pengujian terhadap performansi solusi

A. Implementasi Solusi

Database peta OSM diunduh dari halaman mapzen.com. Di sini database yang dipilih untuk diunduh adalah Bandung, Indonesia. Dasar pemilihan menggunakan mapzen karena peta dapat diunduh sesuai daerah pengujian. Hal ini sulit dilakukan bila menggunakan fitur ekspor dari OSM.

Database yang telah diunduh diekspor ke dalam database Postgis dengan menggunakan osm2pgsql suatu program konverter dari format pbf menuju Postgis. Postgis merupakan database spatial ekstensi dari program Postgres. PostGis digunakan untuk menyimpan data dari format pbf tersebut dalam bentuk *point*, *line*, *multipolygon*.

Dari database Postgis tersebut dapat digambarkan kedalam program bernama Qgis. Proses ini perlu dilakukan karena postgis dapat menyimpan keseluruhan tag dari file osm tersebut. Kemudian setelah konversi dilakukan, digunakan program Qgis untuk menggambar peta yang telah ada.



Gambar 8. Tampilan peta OSM pada QGIS

Pada Gambar 8 peta yang ditampilkan merupakan peta jalanan Bandung untuk kebutuhan routing. Layer yang lain dapat ditampilkan bila dibutuhkan seperti point yang berisi lokasi-lokasi tempat pada peta.

Program pencarian rute terpendek dibuat dalam bahasa C++. Penelusuran pada titik yang diberikan nama menghasilkan solusi rute terpendek. Dimana dapat dilihat pada gambar IV.4. Pengujian perangkat lunak terdiri dari dua bagian, pengujian kebenaran algoritma dan pengujian kinerja perangkat lunak.

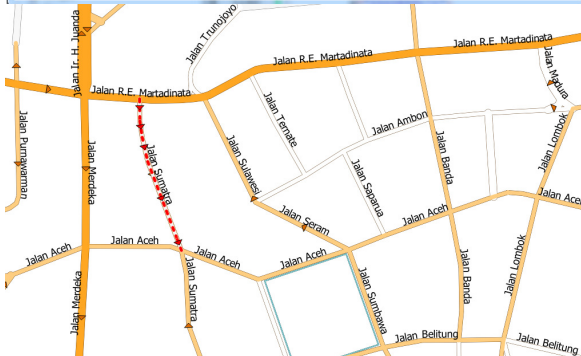
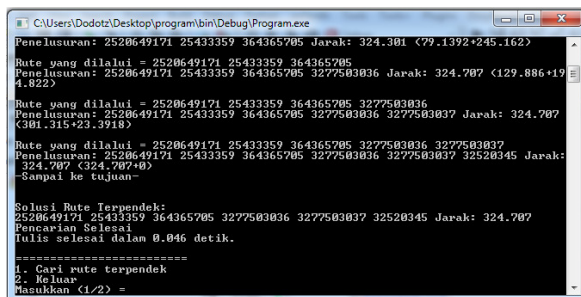
B. Pengujian

Data pengujian perangkat lunak pencarian rute terpendek berasal dari peta petabandung.osm yaitu daerah Bandung yang memiliki batas lintang -6.9116 ke -6.8865 serta bujur 107.5944 ke 107.6380. Ukuran file xml sebesar 4 mb dan memiliki jumlah node sebesar 15000.

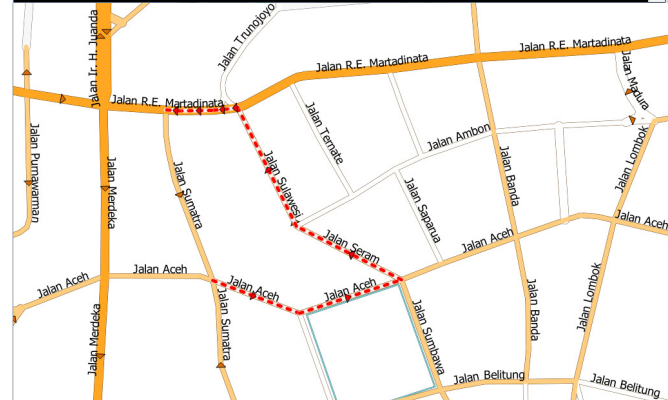
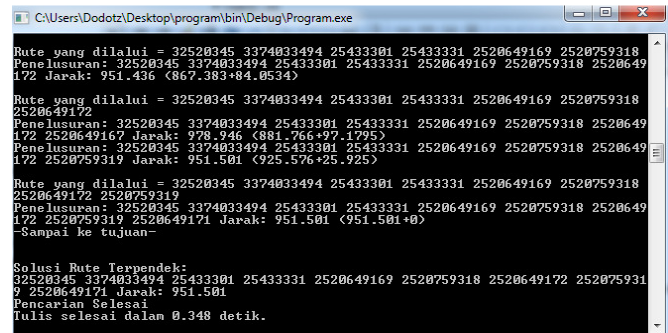


Gambar 9 Kasus uji satu jalur

Kasus uji satu jalur dilakukan pada gambar 8 dimana hendak dilakukan pencarian rute terpendek dari A menuju B dan juga kebalikannya dari B menuju A. Hasil pengujian yang diharapkan adalah A - B berbeda dengan B - A karena algoritma memperhitungkan jalan satu arah.



Gambar 10. Hasil uji arah jalan A – B Hasil pengujian didapatkan pencarian rute ditemukan dalam 0.046 detik. Serta hasil pada peta seperti pada gambar 9 Terlihat rute yang terbentuk lurus melalui Jalan Sumatra.



Gambar 11 Hasil uji arah jalan B - A

Hasil pengujian didapatkan pencarian rute ditemukan dalam 0.348 detik. Serta hasil pada peta seperti pada gambar 10. Dapat dilihat pada peta bahwa hasil B-A melalui jalur lain yaitu melewati Jalan Aceh, Jalan Serang, Jalan Sulawesi, dan Jalan R.E Martadinata.

V. SIMPULAN DAN SARAN

Berikut adalah simpulan dari Tugas Akhir:

1. Algoritma A* dapat diimplementasikan pada OSM dengan cara memarsing file OSM berupa koordinat serta keterhubungan dari xml node dan way. Koordinat tersebut digunakan untuk dibuat graf berarah. Graf tersebut digunakan untuk mencari rute terpendek dari titik awal dan titik akhir yang digunakan.
2. Algoritma A* dapat diujikan pada OSM melalui perencanaan kasus uji yang telah dijabarkan pada bab IV. Algoritma A* diuji dengan kasus yang dapat menyatakan kebenaran serta performansinya.
3. Penggunaan QGIS sebagai perangkat visualisasi berhasil untuk menunjukkan rute perjalanan terpendek dari keluaran program.

4. Berdasarkan hasil eksperimen OpenStreetMap sebagai penyedia layanan peta untuk pencarian rute terpendek dapat digunakan.

Berikut adalah saran dari Tugas Akhir:

1. Peningkatan performansi algoritma dapat dilakukan dengan melakukan proses awal terhadap file osm. Agar tidak semua data perlu diparsing.
2. Data yang telah diproses dapat disimpan pada file eksternal sehingga tidak perlu dilakukan perhitungan berulang. Hal ini dapat dilakukan karena lokasi peta dibatasi hanya daerah Bandung.
3. Pembentukan tipe pengguna pencarian rute agar jalanan yang digunakan sesuai dengan pengguna. Misal: jalanan utama untuk kendaraan dan mobil namun kurang diperhitungkan untuk pejalan kaki atau pesepeda.

REFERENCES

- [1] E. Dijkstra. *A note on two problems in connexion with graphs*, in *Numerische Mathematik*, vol. 1. 1959, halaman. 269–271.
- [2] F. Hanny. *Optimasi Pemilihan Rute Terpendek Jalur Angkutan Umum sesuai Preferensi Pengguna dengan Algoritma A* berbasis Google Maps*. 2013. Bandung: Institut Teknologi Bandung.
- [3] I. Frank. *Calculating Geographic Distance: Concepts and Methods*. 2006. Canadian Institute for Health Information, Toronto, Ontario, Canada
- [4] H. Kaindl and G. Kainz. *Bidirectional heuristic search reconsidered*, in *Journal of Artificial Intelligence Research*, vol. 7, 1997, halaman. 283-317.
- [5] L.K. Masayu. *Route/Path Planning*. Slide kuliah IF3051. ITB
- [6] M. Rinaldi. *Diklat kuliah Matematika Diskrit*. 2002.
- [7] P. Sanders and D. Schultes. *Engineering highway hierarchies*. Proceedings of the 14th European Symposium on Algorithms, 2006, pp. 804–816.
- [8] Pepping, Wouter. *The Shortest Route Developing an online bicycle route planner for the Netherlands*. BMI Paper. 2009.
- [9] R. E. Korf. *Depth-first iterative-deepening: an optimal admissible tree search*. *Artificial Intelligence*, vol. 27, 1985, halaman. 97-109.
- [10] S. Russell and J. Norvig, *Artificial Intelligence A Modern Approach*. Prentice Hall, 1995, halaman. 95-96.
- [11] S. Russell. *Efficient memory-bounded search methods*, in *Proceedings of the 10th European Conference on Artificial intelligence*, 1992, halaman. 1-5.
- [12] <http://www.bsierad.com/measure-the-distance-on-google-map-using-euclidean-and-haversine/> waktu akses 11 Desember 2015 pukul 13.40
- [13] <http://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/> waktu akses 11 Desember 2015 pukul 13.50
- [14] <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html> waktu akses 10 Desember 2014 pukul 11.20.