

Penerapan Algoritma A*(STAR) Untuk Mencari Rute Tercepat Pada Suatu Bengkel

(Studi Kasus di daerah Kebayoran Baru, Jakarta Selatan)

Anggi Kurniawati
Teknik Informatika
Universitas Al Azhar Indonesia
Jakarta Selatan, DKI Jakarta
Email : anggikurniawati@if.uai.ac.id

Abstract—Untuk menempuh perjalanan dari satu tempat ke tempat lain, peralatan yang paling sering digunakan adalah peta. Dengan menggunakan peta dapat diestimasi rute yang optimum dengan menghitung panjang jalan yang akan ditempuh. Saat ini masih banyak yang belum dapat menggunakan pencarian bengkel online yang memudahkan pengguna untuk memperbaiki kendaraan mereka. Oleh karena itu dibutuhkan sistem yang dapat menunjukkan lokasi bengkel beserta jalur terpendeknya, agar waktu pencarian lebih efisien. Terdapat beberapa algoritma pencarian jalur terpendek, salah satunya adalah algoritma A* (A-Star). Algoritma A* menggunakan estimasi jarak terdekat untuk mencapai tujuan (goal) dan memiliki nilai heuristik yang digunakan sebagai dasar pertimbangan. Heuristik adalah kriteria, metoda, atau prinsip-prinsip untuk menentukan pilihan sejumlah alternatif untuk mencapai sasaran dengan efektif.

Keywords—component; pencarian rute, A*, bengkel.

I. PENDAHULUAN

Semakin padatnya pengguna kendaraan mobil dan motor di daerah Kebayoran Baru, Jakarta Selatan. Kita sering melihat orang yang hanya bisa menggunakan kendaraannya saja sehingga jika kendaraan mereka mengalami kerusakan yang mendasak. Kondisi ini membuang waktu anda beraktivitas dan kebutuhan ini mempengaruhi pengguna kendaraan untuk dapat mengetahui lokasi bengkel terdekat secara cepat. Sehingga para pengguna kendaraan juga akan semakin terbantu jika penunjuk lokasi tersebut dapat diakses dengan mudah.

Mengetahui semakin banyak pengguna kendaraan pada saat ini membuat penulis ingin menuaikan suatu ide baru dimana penulis mengidentifikasi suatu lokasi bengkel yang berada di Kebayoran Baru, Jakarta Selatan.

Dalam penelitian sebelumnya penulis belum menentukan jarak bengkel terdekat. Penulis mengidentifikasi bengkel besar hingga tambal ban di pinggir jalan. Untuk solusi saat ini penulis ingin membuat jarak terdekat dari lokasi dan memudahkan pengguna. Agar dapat menemukan solusi terbaik dalam pencarian bengkel terdekat yang memiliki kompleksitas yang rendah maupun yang tinggi, digunakan suatu algoritma pencarian. Dalam penyelesaian ini, akan digunakan Algoritma A* dimana algoritma ini merupakan salah satu algoritma *path finding* yang sering digunakan.

II. TINJAUAN PUSTAKA

2.1 Graph

Graph digunakan untuk mempresentasikan objek-objek dan hubungan antara objek-objek tersebut. Sebuah graph (G) dinyatakan sebagai pasangan tupel [1] :

$$G = (V, E)$$

Keterangan :

V : himpunan berhingga verteks/simpul

E : himpunan berhingga edge.

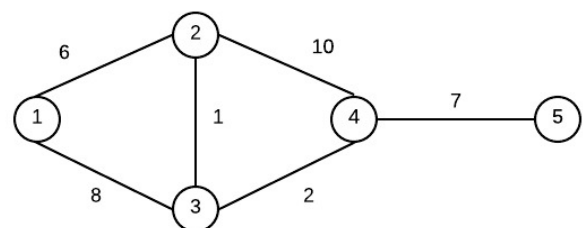
Jika sebuah edge (v1, v2) terdapat di E maka v1 dan v2 terdapat di V. Sebuah graph disebut berbobot jika setiap edge (v1, v2) memiliki nilai yang disebut sebagai bobot. Sebuah edge dalam graph berbobot dinyatakan dalam bentuk (v1, v2, k) dengan v1 dan v2: verteks di V.

K: bobot

Sebuah *graph* dapat dinyatakan dalam bentuk gambar dengan lingkaran menyatakan verteks dan busur yang menghubungkan lingkaran menyatakan sisi. Gambar1 adalah contoh sebuah *graph* berbobot. Jika dinyatakan dalam bentuk $G=(V,E)$ maka :

$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{(1, 2, 6), (1, 3, 8), (2, 3, 1), (2, 4, 10), (3, 4, 2), (4, 5, 7)\}$$



Gambar1. Contoh graph berbobot

2.3 Algoritma A*

Algoritma A* merupakan contoh dari best-first search. Penggunaan algoritma ini dengan fungsi heuristik yang tepat dapat memberikan hasil yang optimal, maka algoritma ini pun disebut A*[2].

Beberapa terminologi dasar yang terdapat pada algoritma ini :

- starting point
- simpul
- open list
- closed list
- harga (*cost*)
- halangan (*unwalkable*)

Pada kondisi yang tepat, Algoritma A* akan memberikan solusi yang terbaik dalam waktu yang optimal. Algoritma ini memeriksa simpul dengan menggabungkan $g(n)$, yaitu *cost* yang dibutuhkan untuk mencapai sebuah simpul dan $h(n)$ yaitu *cost* yang didapat dari simpul ke tujuan [2].

Sehingga dapat dirumuskan sebagai berikut :

$$f(n) = g(n) + h(n)$$

Dimana:

- $f(n)$: Fungsi evaluasi
- $g(n)$: Jarak total dari posisi asal ke lokasi sekarang
- $h(n)$: Fungsi heuristik yang digunakan untuk memperkirakan jarak dari lokasi sekarang ke lokasi tujuan. Semakin tinggi keakuratan heuristik, semakin cepat dan bagus lokasi tujuan ditemukan dan dengan tingkat keakuratan yang lebih baik.

Untuk mencapai simpul berikutnya didapat dari fungsi $f(n)$, sehingga pada pemilihan jalur terpendek dapat langsung diketahui simpul berikutnya dengan *cost* terkecil sampai mencapai simpul tujuan tanpa kembali ke simpul yang sudah dikunjungi.

Pada Algoritma A*, dibutuhkan 2 antrian yaitu:

- OPEN, yang berisi simpul-simpul yang sudah dibangkitkan, sudah memiliki fungsi heuristik namun belum diuji.
- CLOSED, berisi simpul-simpul yang sudah diuji.

Algoritma A* secara garis besar dapat dijelaskan seperti berikut:

- Masukkan simpul awal ke *open list*
- Loop langkah-langkah dibawah ini:
 - Cari simpul (n) dengan nilai $f(n)$ yang paling kecil dalam *open list*, dan simpul ini sekarang menjadi *current* simpul.
 - Keluarkan *current* simpul dari *open list* dan masukkan ke *open list*
 - Untuk setiap tetangga dari *current* simpul lakukan berikut:
 - Jika tidak dapat dilalui atau sudah ada dalam *close*

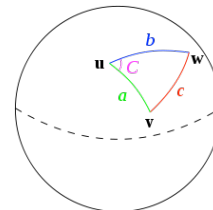
list, abaikan

- Jika belum ada di *open list* buat *current* simpul *parent* dari simpul tetangga ini, simpan nilai f , g , dan h dari simpul ini.
- Jika sudah ada di *open list* cek apabila simpul tetangga ini lebih baik menggunakan nilai g sebagai ukuran. Jika lebih baik ganti *parent* dari simpul ini di *open list* menjadi *current* simpul, lalu kalkulasikan ulang nilai g dan f dari simpul ini.
- Hentikan *looping* jika:
 - Simpul tujuan telah ditambah ke *open list* yang berarti rute ditemukan
 - Belum menemukan simpul akhir (tujuan) sementara *open list* kosong atau berarti tidak ada rute.

Simpan rute, lalu secara *backward* urut mulai dari simpul akhir (tujuan) sampai ke titik awal sambil menyimpan simpul ke dalam array.

2.3 Haversine Formula

Haversine formula merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan[4].



Gambar2. Ilustrasi Haversine Formula

$$\cos(c) = \cos(\alpha)\cos(b) + \sin(\alpha)\sin(b)\cos(c) \quad (1)$$

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

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

Gambar3. Rumus Busur

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

Penggunaan rumus ini mengasumsikan pengabaian efek *ellipsoidal*, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi :

$$d = 2r \cdot \arcsin\left\{\sqrt{\sin^2\left(\frac{Lat_1 - Lat_2}{2}\right) + \cos(lat_1)\cos(lat_2)\sin^2\left(\frac{Long_1 - Long_2}{2}\right)}\right\}$$

keterangan :

- d : Jarak
- Lat₁ : Latitude Awal
- Lat₂ : Latitude Akhir
- Long₁ : Longitude Awal
- Long₂ : Longitude Akhir
- r : Radius Bumi = 6371km
- 1 derajat : 0.0174532925 radian

III. HASIL DAN PEMBAHASAN

Dengan pembahasan menggunakan A* untuk pencarian jalur terdekat untuk mencapai bengkel terdekat. Pada kali ini implementasi dengan menggunakan rumus yang sudah ada.

Untuk pencarian bengkel terdekat diperlukan lokasi titik awal yang nantinya digunakan untuk implementasi metode *haversine formula*. Pada penelitian ini lokasi titik awal adalah **Universitas Al Azhar Indonesia**. Pengumpulan data disertakan alamat lengkap dari lokasi akhir :

No	Bengkel	Alamat
1	Yanuar	Jl. Kyai Maja B1 No.0/35
2	Permaisuri	Jl. Mahakam No.5 RT.1/RW.6
3	IMS Indonesia	Jl. Melawai Raya No.193C
4	Daman Motor	Jl. Gandaria II No. 3

Berikut data tabel lokasi awal dan lokasi akhir :

No	Lokasi Awal	Lattitude	Longitude
1	Universitas Al Azhar Indonesia	-6.236248	106.799189

1. Menentukan lokasi awal pencarian
2. Menentukan lokasi akhir pencarian

No	Bengkel	Lattitude	Longitude
1	Yanuar	-6.241407	106.793316
2	Permaisuri	-6.244204	106.796493
3	IMS Indonesia	-6.246059	106.797567
4	Daman Motor	-6.243546	106.788292

Hasil perhitungan nilai heuristiknya dengan menggunakan metode *haversine* :

1. Dari Universitas Al Azhar ke Bengkel Yanuar
 - b. Lokasi awal
 - Lattitude : -6.236248
 - Longitude : 106.799189

- b. Lokasi Tujuan
 - Lattitude : -6.241407
 - Longitude : 106.793316

Heuristik = 0.000002 km
= 0.002 m

2. Dari Universitas Al Azhar ke Bengkel Permaisuri

- a. Lokasi awal
 - Lattitude : -6.236248
 - Longitude : 106.799189

- b. Lokasi Tujuan
 - Lattitude : -6.244204
 - Longitude : 106.796493

Heuristik = 0.000003 km
= 0.003 m

3. Dari Universitas Al Azhar ke Bengkel IMS Indonesia

- a. Lokasi awal
 - Lattitude : -6.236248
 - Longitude : 106.799189

- b. Lokasi Tujuan
 - Lattitude : -6.246059
 - Longitude : 106.797567

Heuristik = 0.000003 km
= 0.003 m

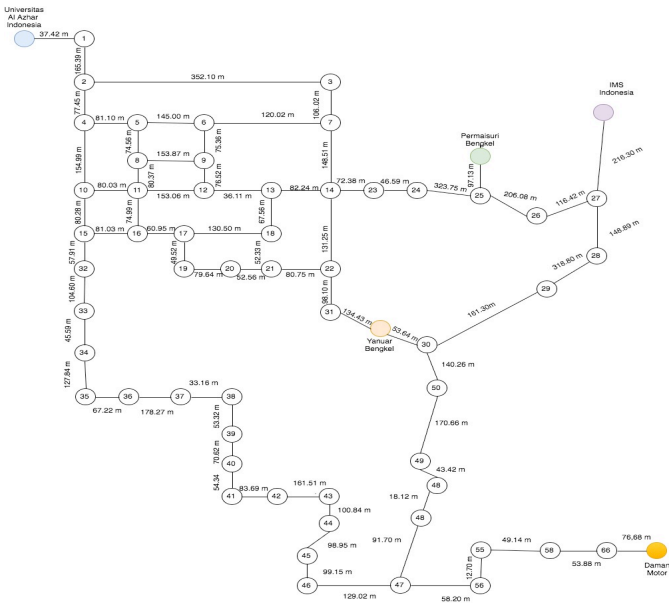
4. Dari Universitas Al Azhar ke Bengkel Daman Motor

- a. Lokasi awal
 - Lattitude : -6.236248
 - Longitude : 106.799189

- b. Lokasi Tujuan
 - Lattitude : -6.243546
 - Longitude : 106.788292

Heuristik = 0.000004 km
= 0.004 m

Setelah menemukan nilai *heuristik* maka menentukan nilai *g(n)* atau jarak total dari lokasi awal dari posisi awal ke posisi akhir dengan menggunakan fitur "*measure distance*" dari *google maps*. Dengan menggunakan *measure distance* dapat membuat simpul sehingga dapat mengetahui jarak tiap simpulnya, yaitu dari persimpangan satu ke persimpangan yang lain. Dan dapat diukur dengan menggunakan jalur jalan rayanya, bukan garis lurus. Berikut ini adalah gambar yang terdapat simpul sesuai dengan keadaan jalan raya empat bengkel terdekat dari Universitas Al Azhar Indonesia di daerah kebayoran baru.



Gambar4. Contoh Simpul

Perhitungan dengan menggunakan Algoritma A* untuk mencari bengkel terdekat dari lokasi awal yaitu Yanuar Bengkel :

$$f(n) = g(n) + h(n)$$

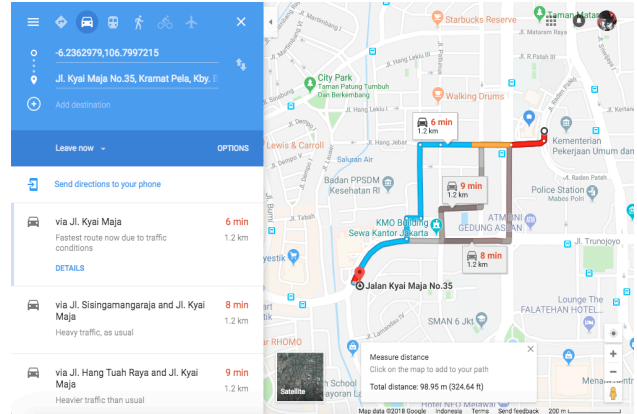
1. Simpul 1
 $f(n) = g(n) + h(n)$
 $= 37,44 + 0,02$
 $= 37,44$ m
2. Simpul 2
 $f(n) = g(n) + h(n)$
 $= 165,39 + 0,02$
 $= 165,41$ m
3. Simpul 4
 $f(n) = g(n) + h(n)$
 $= 77,45 + 0,02$
 $= 77,47$ m
4. Simpul 5
 $f(n) = g(n) + h(n)$
 $= 81,10 + 0,02$
 $= 81,12$ m
5. Simpul 8
 $f(n) = g(n) + h(n)$
 $= 74,56 + 0,02$
 $= 74,58$ m
6. Simpul 11
 $f(n) = g(n) + h(n)$
 $= 80,37 + 0,02$
 $= 80,39$ m
7. Simpul 16
 $f(n) = g(n) + h(n)$
 $= 74,99 + 0,02$
 $= 75,01$ m
8. Simpul 17
 $f(n) = g(n) + h(n)$
 $= 60,95 + 0,02$

$$= 60,97$$
 m

9. Simpul 19
 $f(n) = g(n) + h(n)$
 $= 49,52 + 0,02$
 $= 49,54$ m
10. Simpul 20
 $f(n) = g(n) + h(n)$
 $= 79,64 + 0,02$
 $= 79,66$ m
11. Simpul 21
 $f(n) = g(n) + h(n)$
 $= 52,56 + 0,02$
 $= 52,58$ m
12. Simpul 22
 $f(n) = g(n) + h(n)$
 $= 80,75 + 0,02$
 $= 80,77$ m
13. Simpul 31
 $f(n) = g(n) + h(n)$
 $= 98,10 + 0,02$
 $= 98,12$ m
14. Simpul end
 $f(n) = g(n) + h(n)$
 $= 134,43 + 0,02$
 $= 134,45$ m

Total = 1147,51 m
 = 1.14751 km

Hasil dari total perhitungan menggunakan Algoritma A* adalah 1,147 km. Berikut hasil dari Google Maps :



Gambar5. Hasil Dari Google Maps

Setelah dilihat dari hasil perhitungan mencari bengkel terdekat diatas menggunakan algoritma A* dengan Google Maps hasilnya tidak terlalu jauh sehingga Algoritma A* sendiri melihat dari jarak yang terpendek dari setiap simpul. Sehingga melakukan perbandingan apabila dalam simpul tersebut jaraknya terlalu besar daripada yang sebelumnya, maka yang diambil adalah jarak yang terendahnya. Sehingga algoritma A* menghasilkan jalur sendiri. Kalau dibandingkan dengan Google Maps, Google Maps sendiri memiliki 3 jalur untuk

mencapai ke lokasi tujuan, karena google maps sendiri mencari jalur yang tercepat berdasarkan waktu. Sehingga *Google Maps* sendiri tidak terlalu memikirkan seberapa jauh yang ditempuh oleh pengguna. *Google Maps* sendiri memberikan 3 jalur jalan jaraknya sama saja 1,2 km tetapi waktunya berbeda. Tetapi, kalau algoritma A* memiliki jalur sendiri yang tidak dilewati oleh *Google Maps*.

Perhitungan dengan menggunakan Algoritma A* untuk mencari bengkel terjauh dari lokasi awal yaitu bengkel Daman Motor :

$$f(n)=g(n)+h(n)$$

1. Simpul 1

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 37,44+0,04 \\ &= 37,48 \text{ m} \end{aligned}$$

2. Simpul 2

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 165,39+0,04 \\ &= 165,43 \text{ m} \end{aligned}$$

3. Simpul 4

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 77,45+0,04 \\ &= 77,49 \text{ m} \end{aligned}$$

4. Simpul 5

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 81,10+0,04 \\ &= 81,14 \text{ m} \end{aligned}$$

5. Simpul 8

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 74,56+0,04 \\ &= 74,60 \text{ m} \end{aligned}$$

6. Simpul 11

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 80,37+0,04 \\ &= 80,41 \text{ m} \end{aligned}$$

7. Simpul 16

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 74,99+0,04 \\ &= 75,03 \text{ m} \end{aligned}$$

8. Simpul 17

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 60,95+0,04 \\ &= 60,99 \text{ m} \end{aligned}$$

9. Simpul 19

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 49,52+0,04 \\ &= 49,56 \text{ m} \end{aligned}$$

10. Simpul 20

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 79,64+0,04 \\ &= 79,68 \text{ m} \end{aligned}$$

11. Simpul 21

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 52,56+0,04 \\ &= 52,60 \text{ m} \end{aligned}$$

12. Simpul 22

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 80,75+0,04 \\ &= 80,79 \text{ m} \end{aligned}$$

13. Simpul 31

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 98,10+0,04 \\ &= 98,14 \text{ m} \end{aligned}$$

14. Simpul yanuar bengkel

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 134,43+0,04 \\ &= 134,47 \text{ m} \end{aligned}$$

15. Simpul 30

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 53,64+0,04 \\ &= 53,68 \text{ m} \end{aligned}$$

16. Simpul 51

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 140,26+0,04 \\ &= 140,30 \text{ m} \end{aligned}$$

17. Simpul 50

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 170,66+0,04 \\ &= 170,70 \text{ m} \end{aligned}$$

18. Simpul 49

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 43,42+0,04 \\ &= 43,46 \text{ m} \end{aligned}$$

19. Simpul 48

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 18,12+0,04 \\ &= 18,16 \text{ m} \end{aligned}$$

20. Simpul 47

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 91,70+0,04 \\ &= 91,74 \text{ m} \end{aligned}$$

21. Simpul 52

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 58,20+0,04 \\ &= 58,24 \text{ m} \end{aligned}$$

22. Simpul 53

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 12,70+0,04 \\ &= 12,74 \text{ m} \end{aligned}$$

23. Simpul 54

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 49,14+0,04 \\ &= 49,18 \text{ m} \end{aligned}$$

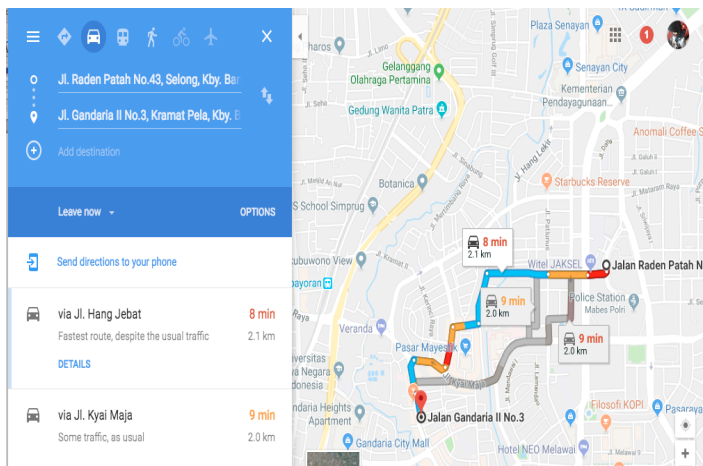
24. Simpul 55

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 53,88+0,04 \\ &= 53,92 \text{ m} \end{aligned}$$

25. Simpul end

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &= 76,68+0,04 \\ &= 76,72 \text{ m} \end{aligned}$$

Total = 1916,55m
= 1.91655 km



Gambar5. Hasil Dari Google Maps

Adapun hasil pencarian bengkel terjauh menggunakan algoritma A* berdasarkan perhitungan langkah-langkah diatas dapat dinyatakan dengan pencarian menggunakan algoritma A* memiliki jalur sendiri, sehingga mendapatkan hasil 1,9 km yang tidak terlalu jauh dengan *Google Maps* yaitu 2,1 km. Seperti perhitungan yang terdekat *Google Maps* memberikan 3 jalur yaitu ada yang 2,0 km dan ada 2,1 km. Kalau yang 2,0 terlihat ada kemacetan sehingga waktunya berbeda dengan yang 2,1 km.

IV. KESIMPULAN

Rancangan yang ditampilkan dalam makalah ini diharapkan dapat diimplementasikan. Meskipun begitu perlu tinjauan lebih lanjut agar sistem dapat berjalan lebih efisien. Ada kemungkinan bahwa sebagian verteks-verteks dari graph tidak akan digunakan dalam perhitungan karena terlalu jauh dari jalur yang akan dicari. Apalagi jika jalur yang dicari sebenarnya relatif cukup dekat dibandingkan dengan luasnya peta.

Hal ini akan mengurangi efisiensi karena pembentukan graph juga memerlukan waktu. Efisiensi juga menjadi masalah jika setiap dilakukan pencarian jalur, graph selalu dibentuk dari peta. Lebih baik jika graph dibentuk saat terdapat perubahan terhadap peta.

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. Untuk menentukan jalur terpendek pada pencarian dengan menggunakan algoritma A*, dibutuhkan beberapa data yaitu, jarak sebenarnya antara simpul yang berhubungan (cost antara simpul). Dengan data tersebut, maka pencarian jalur terpendek dapat diimplementasikan.

2. Dalam proses implementasinya algoritma A*, hasilnya agak sedikit berbeda dari *google maps*. Jadi solusi dari penerapan algoritma A* sendiri pada penelitian ini menghitung secara manual dengan data-data yang sudah didapat kemudian di dibuatkan garis seperti hasil perhitungan pada rute *google maps*.

REFERENCES

- [1] Munir, R., 2014. "Matematika Diskrit", Edisi kelima.
- [2] Russell, Stuart. & Norvig, Peter. 2003. "Artificial Intelligence: A Modern Approach", New Jersey: Prentice Hall.
- [3] Chopde, Nitin R., & Nichat, Mangesh K. 2013. Landmark Based Shortest 6 Path Detection by Using A* and Haversine Formula. International Journal of Innovative Research in Computer and Communication Engineering