

Penerapan Algoritma A Menggunakan Graf untuk Menentukan Rute Terpendek dari Jalur Alternatif Purworejo - Jogja*

Ferdina Wiranti Afifah 13518046
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518046@std.stei.itb.ac.id

Abstract—Algoritma A* merupakan salah satu algoritma untuk menentukan rute terpendek. Algoritma tersebut menghasilkan rute terpendek yang efisien berdasarkan jarak yang ditempuh dan straight-line distance yang merupakan jarak lurus dari suatu titik kota menuju kota tujuannya. Proses penentuan rute terpendek dari jalur alternatif Purworejo-Jogja yang menggunakan algoritma A* sesuai dengan informasi yang ada pada *google map* sehingga hal tersebut membuktikan bahwa algoritma A* merupakan algoritma yang tepat untuk menentukan rute terpendek dari beberapa jalur.

Keywords—algoritma; A*; rute terpendek; jalur alternatif; Purworejo; Jogja; pertimbangan.

I. PENDAHULUAN

Purworejo merupakan salah satu kabupaten yang berada di Jawa Tengah, yang mana kabupaten tersebut adalah tempat tinggal penulis saat ini. Sering kali penulis jalan-jalan bersama keluarga ke tempat wisata yang berada di Kota Jogja karena jaraknya cukup dekat dengan tempat tinggal kami sehingga tidak menghabiskan waktu di jalan jika waktu libur yang kami miliki sangat singkat. Ada banyak jalur alternatif yang dapat dilalui dari Purworejo untuk menuju Kota Jogja. Keluarga penulis telah melalui banyak jalur untuk mendapatkan pengalaman perjalanan yang berbeda. Ada jalur yang jalannya naik turun, ada juga yang jalannya benlubang, lalu ada jalur dengan pemandangan yang indah di kanan kiri jalan, dan lain sebagainya.

Ketika akan melakukan perjalanan dari suatu tempat ke tempat lain pasti sangat mempertimbangkan berbagai hal, seperti efisiensi biaya dan waktu, maka diperlukan pengetahuan untuk menentukan rute terpendek. Algoritma pencarian rute terpendek atau lebih dikenal sebagai *shortest-path* dipakai dalam menentukan rute dalam sebuah graf [5].

Teori graf merupakan cabang kajian dalam matematika dan ilmu komputer. Graf digunakan untuk merepresentasikan sebuah objek yang memiliki keterhubungan dengan objek lain. Banyak hal yang dapat diterapkan menggunakan graf, salah satunya adalah penentuan rute terpendek dari beberapa jalur alternatif.

Teknik pencarian yang sering digunakan untuk menentukan rute terpendek yaitu pencarian buta (*blind search*) dan pencarian heuristik (*heuristic search*). Pencarian buta

cenderung lebih mudah dipahami dibandingkan pencarian heuristik, tetapi hasil pencarian yang diperoleh pencarian heuristik lebih variatif dan waktu pencarian solusi lebih cepat [5].

Ada beberapa cara mencari optimasi rute terpendek untuk menghubungkan satu tempat dengan tempat yang lain diantaranya adalah algoritma Dijkstra, algoritma Semut atau *Ant Colony*, algoritma Floyd Warshall, algoritma Bellman Ford, algoritma Distance Vector, algoritma Ford Fulkerson, dan algoritma A*. Algoritma tersebut dibuat untuk mencari cara yang paling efisien dan efektif untuk menganalisa pemilihan jalur tercepat dengan berbagai penambahan variable [5].

Salah satu metode pencarian jalur terpendek yang termasuk dalam pencarian heuristik adalah Algoritma A*. Algoritma A* merupakan salah satu algoritma pencarian rute yang optimal dan komplit. Optimal berarti rute yang dihasilkan adalah rute yang paling baik dan komplit berarti algoritma tersebut dapat mencapai tujuan yang di harapkan. Dalam penerapannya, Algoritma A* menggunakan jarak sebagai proses kalkulasi nilai terbaik [5].

Namun, jika ingin melakukan suatu perjalanan, selain memilih jalur dengan rute terpendek untuk menghemat waktu dan biaya, hendaknya mempertimbangkan sesuatu (rintangan, hambatan, dan kondisi lain) yang ada pada jalur yang akan dilalui tersebut agar tidak kaget saat melaluinya. Pada makalah ini penulis berikan informasi mengenai pertimbangan yang harus diperhatikan saat memilih rute terbaik dari jalur alternatif Purworejo-Jogja yang telah penulis pilih.

II. DASAR TEORI

A. Graf

Graph adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*) (Zakaria, 2006). Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E .

1. Verteks (simpul), yaitu himpunan simpul yang terbatas dan tidak kosong.
2. Edge (sisi/busur), yaitu himpunan busur yang menghubungkan sepasang simpul [4].

B. Lintasan Terpendek

Lintasan terpendek (*shortest path*) merupakan lintasan minimum yang diperlukan untuk mencapai suatu titik dari titik tertentu. Dalam pencarian lintasan terpendek masalah yang dihadapi adalah mencari lintasan mana yang akan dilalui sehingga didapat lintasan yang paling pendek dari satu verteks ke verteks yang lain. Ada beberapa macam persoalan lintasan terpendek, antara lain : (Pawitri, dkk., 2007)

- Lintasan terpendek antara dua buah verteks.
- Lintasan terpendek antara semua pasangan verteks.
- Lintasan terpendek dari verteks tertentu ke semua verteks yang lain.
- Lintasan terpendek antara dua buah verteks yang melalui beberapa verteks tertentu [4].

C. Algoritma A*

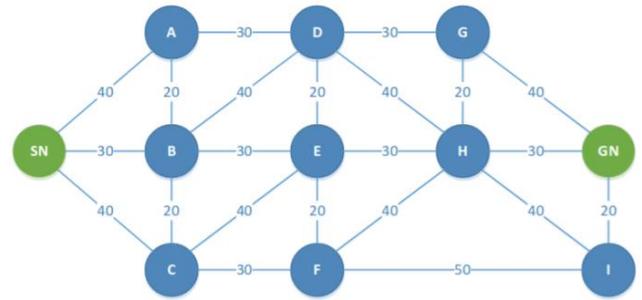
Pada tahun 1967, Bertram Raphael membuat perbaikan-perbaikan dramatis atas algoritma ini, tapi gagal menunjukkan keoptimasiannya. Ia menyebut algoritma ini A2. Kemudian pada tahun 1968 Peter E. Hart memperkenalkan sebuah argumen yang membuktikan A2 optimal ketika menggunakan heuristik konsisten hanya dengan perubahan kecil. Pembuktiannya atas algoritma tersebut juga termasuk bagian yang menunjukkan bahwa algoritma A2 yang baru adalah algoritma terbaik. Maka dari itu, ia menamai algoritma baru ini dalam sintaksis Kleene Star untuk menjadi algoritma yang berawal dari A dan memasukkan semua nomor-nomor yang mungkin dari A*. Dalam sains komputer, A* (dibaca "A star") adalah algoritma komputer yang digunakan secara luas dalam mencari jalur (pathfinding) dan grafik melintang (graph traversal), proses plotting sebuah jalur melintang secara efisien antara titik-titik, disebut node (Suyanto, 2014) [1].

Algoritma A* merupakan perbaikan dari metode best first search dengan memodifikasi fungsi heuristiknya, Algoritma A* akan meminimumkan total biaya lintasan. Pada kondisi yang tepat, Algoritma A* akan memberikan solusi yang terbaik dalam waktu yang optimal (Kusumadewi, 2003) [4].

Algoritma A Star atau A* adalah salah satu algoritma pencarian yang menganalisa input, mengevaluasi sejumlah jalur yang mungkin dilewati dan menghasilkan solusi. Algoritma A* adalah algoritma komputer yang digunakan secara luas dalam graph traversal dan penemuan jalur serta proses perencanaan jalur yang bisa dilewati secara efisien di sekitar titik-titik yang disebut node (Reddy, 2013) [1].

Karakteristik yang menjelaskan algoritma A* adalah pengembangan dari "daftar tertutup" untuk merekam area yang dievaluasi. Daftar tertutup ini adalah sebuah daftar untuk merekam area berdekatan yang sudah dievaluasi, kemudian melakukan perhitungan jarak yang dikunjungi dari "titik awal" dengan jarak diperkirakan ke "titik tujuan" (Reddy, 2013) [1].

Pada tahun 1964 Nils Nilsson menemukan heuristik berdasarkan pendekatan untuk menambah kecepatan pada Algoritma Dijkstra. Algoritma ini disebut dengan A1.



Gambar 1. Graf Dijkstra

Pada gambar 1, suatu graph dengan Node Awal SN dan Node Tujuan GN terhubung dengan node - node lain oleh edge yang memiliki besaran yang berbeda. Jika ditelusuri, terdapat banyak kombinasi rute yang dapat dilalui untuk menuju node tujuan. Bisa dikatakan dari graph tersebut, setiap node akan memberikan solusi arah menuju node tujuan. A* merupakan bentuk yang paling dari Best First Search (BFS). A* mengevaluasi node dengan menggabungkan $g(n)$, yaitu cost untuk mencapai node, dan $h(n)$, yaitu cost yang diperlukan dari node untuk mencapai tujuan, dalam notasi matematika dituliskan sebagai:

$$f(n) = g(n) + h(n), \text{ dengan}$$

$f(n)$ adalah biaya evaluasi;

$g(n)$ adalah biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan n ;

$h(n)$ adalah estimasi biaya untuk sampai pada suatu tujuan mulai dari n [1].

D. Fungsi Heuristik

Kata heuristic berasal dari sebuah kata kerja bahasa Yunani, heuriskein, yang berarti "mencari atau menemukan". Dalam dunia pemrograman, sebagian orang menggunakan kata heuristic sebagai lawan kata dari algoritma, dimana kata heuristic ini diartikan sebagai suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditentukan. Didalam mempelajari metode-metode pencarian ini, kata heuristic diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi. Teknik pencarian heuristic (heuristic searching) merupakan suatu strategi untuk melakukan proses pencarian secara selektif dan dapat memandu proses pencarian yang memiliki kemungkinan sukses paling besar, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Untuk menerapkan pencarian heuristic diperlukan suatu fungsi heuristic. Fungsi heuristic adalah aturan-aturan yang digunakan untuk mendapatkan solusi yang diinginkan. Heuristik jarak ditambah biaya adalah penjumlahan dari dua fungsi:

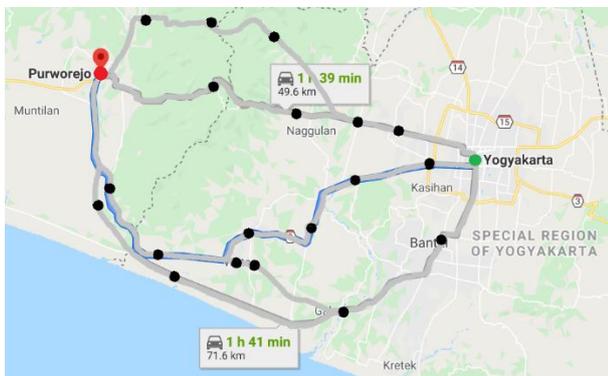
1. Fungsi jalur biaya, yang mana biayanya dihitung dari node awal hingga node saat ini (biasanya dinotasikan $g(n)$).
2. Estimasi heuristik yang dapat diterima (*admissible*) dari jarak menuju goal (biasanya dinotasikan $h(n)$) [1].

III. ANALISIS MASALAH

A. Visualisasi Graf pada Jalur Alternatif Purworejo-Jogja untuk Penerapan Algoritma A*

Graf digunakan untuk memudahkan penulis dalam menentukan rute terpendek dari beberapa jalur alternatif dari Kabupaten Purworejo menuju Kota Jogja. Graf tersebut sebagai visualisasi dari rute yang penulis dapatkan dari *google map*. Titik-titik (simpul) yang terdapat pada graf merupakan daerah yang telah penulis tentukan dari hasil diskusi penulis dengan bapak penulis, yang mana bapak dari penulis sudah paham betul dengan berbagai jalur alternatif menuju Jogja.

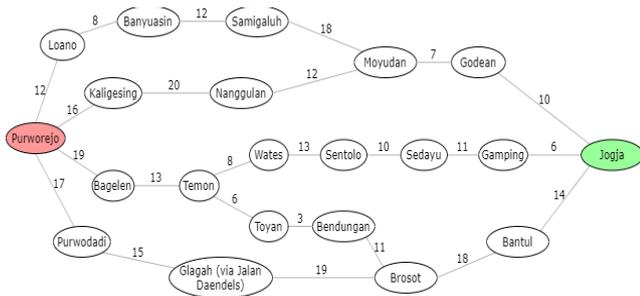
Berikut adalah gambaran jalur alternatif dari Purworejo menuju Jogja yang penulis ambil dari *google map* dengan hasil modifikasi dari penulis.



Gambar 2. Visualisasi jalur alternatif Purworejo-Jogja dari *google map*

Pada gambar 2 terlihat bahwa terdapat beberapa garis berwarna abu-abu yang merupakan penggambaran dari beberapa jalur yang dapat dilalui menuju Jogja. Selain itu, terdapat titik-titik diatas garis abu-abu yang menggambarkan daerah tertentu yang telah penulis tentukan sebagai patokan.

Berikut adalah graf yang menunjukkan jalur alternatif dari Purworejo menuju Jogja sebagai hasil penyederhanaan dari penggambaran dari *google map* beserta jarak tempuh antar daerah.



Gambar 3. Graf dari jalur alternatif Purworejo-Jogja

Pada gambar 3 terlihat graf yang menunjukkan hubungan-hubungan antar daerah yang akan dilalui ketika hendak ke Jogja dari Purworejo. Gambar tersebut juga memperlihatkan bahwa banyak alternatif jalan yang dapat dilalui menuju Jogja.

Dari visualisasi graf hasil penyederhanaan dari peta di *google map*, penulis dapat menentukan rute terpendek dari beberapa jalur alternatif Purworejo-Jogja dengan mudah. Graf tersebut akan dijadikan acuan untuk menyelesaikan persoalan tersebut menggunakan algoritma A*.

B. Penerapan Algoritma A* dalam Menentukan Rute Terpendek dari Jalur Alternatif Purworejo-Jogja

Algoritma A* memang banyak digunakan untuk menentukan rute terpendek suatu tempat yang menghubungkan tempat lain. Hal tersebut dikarenakan dalam prosesnya, algoritma A* akan mengevaluasi beberapa jalur yang mungkin dilewati dan akan menghasilkan solusi dengan rute terpendek yang dihasilkan dari penjumlahan *cost* dan fungsi heuristiknya.

Pada persoalan dalam makalah ini, untuk menentukan rute terpendek dengan algoritma A* digunakan rumus fungsi evaluasi sebagai berikut.

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

$g(n)$ adalah jarak tempuh antar daerah yang ditunjukkan dengan angka-angka yang terdapat pada sisi antar simpul dalam graf. Lalu $h(n)$ adalah fungsi heuristik, yang mana merupakan *straight-line distance* (jarak secara garis lurus) antara daerah-daerah patokan dengan Jogja yang telah penulis dapatkan dari *google map* menggunakan fasilitas (fitur) *measure distance* yang telah disediakan oleh *google map*.

Berikut adalah tabel *straight-line distance* tiap daerah menuju Jogja (dalam km).

Tabel 1. *Straight-line distance*

Bagelen	39	Moyudan	13
Bantul	10	Nanggulan	18
Banyuasin	33	Purwodadi	41
Bendungan	26	Purworejo	41
Brosot	22	Samigaluh	26
Gamping	5	Sedayu	14
Glagah	35	Sentolo	20
Godean	9	Temon	34
Jogja	0	Toyan	28
Kaligesing	30	Wates	28
Loano	37		

Dalam algoritma A*, untuk mendapatkan solusi ada yang namanya simpul hidup dan simpul ekspan. Simpul hidup adalah simpul yang telah dibangkitkan oleh simpul akar dan simpul ekspan sebelumnya. Simpul ekspan merupakan simpul yang memiliki nilai fungsi evaluasi paling kecil di antara simpul hidup yang lain.

Untuk mendapatkan solusi seperti pada persoalan dalam makalah ini, langkah-langkah yang dapat dilakukan adalah sebagai berikut.

1. Simpul mula-mula dibangkitkan dari simpul akar, yaitu Purworejo, yang mana merupakan titik awal untuk memulai perjalanan menuju Jogja, seperti yang terlihat pada gambar 3.
2. Dari simpul akar tersebut, dibangkitkan empat simpul yang akan menjadi simpul hidup yang mana keempat simpul tersebut memiliki hubungan langsung dengan simpul akar.
3. Kemudian, simpul hidup yang telah dibangkitkan diurutkan berdasarkan fungsi evaluasi yang paling kecil. Lalu, simpul dengan nilai $f(n)$ paling kecil diekspan.
4. Lakukan langkah 3 hingga solusi ditemukan. Setelah mengekspan simpul, jangan lupa untuk membangkitkan simpul hidup yang belum pernah dibangkitkan sebelumnya.
5. Setelah solusi ditemukan, simpul hidup yang tidak diekspan dapat dimatikan untuk menyelesaikan pencairan rute terpendek dengan algoritma A*.

Dengan adanya graf yang menunjukkan beberapa jalur alternatif dari Purworejo menuju Jogja dan *straight-line distance* yang telah diketahui, penulis dapat menentukan rute terpendek menggunakan algoritma A* dengan menggunakan tabel simpul ekspan dan simpul hidup sebagai berikut.

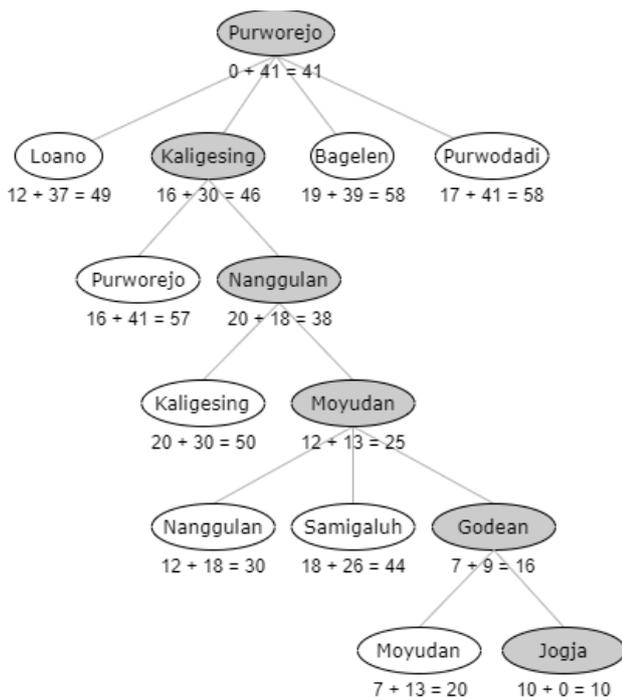
Tabel 2. Simpul ekspan dan simpul hidup

Simpul Ekspan	Simpul Hidup
Purworejo	Kaligesing - Purworejo $f(n) = 16 + 30 = 46$
	Loano - Purworejo $f(n) = 12 + 37 = 49$
	Bagelen - Purworejo $f(n) = 19 + 39 = 58$
	Purwodadi - Purworejo $f(n) = 17 + 41 = 58$
Kaligesing - Purworejo	Nanggulan - Kaligesing - Purworejo $f(n) = 20 + 18 = 38$

	Loano - Purworejo $f(n) = 12 + 37 = 49$
	Bagelen - Purworejo $f(n) = 19 + 39 = 58$
	Purwodadi - Purworejo $f(n) = 17 + 41 = 58$
Nanggulan - Kaligesing - Purworejo	Moyudan - Nanggulan - Kaligesing - Purworejo $f(n) = 12 + 13 = 25$
	Loano - Purworejo $f(n) = 12 + 37 = 49$
	Bagelen - Purworejo $f(n) = 19 + 39 = 58$
	Purwodadi - Purworejo $f(n) = 17 + 41 = 58$
	Godean - Moyudan - Nanggulan - Kaligesing - Purworejo $f(n) = 7 + 9 = 16$
	Samigaluh - Moyudan - Nanggulan - Kaligesing - Purworejo $f(n) = 18 + 26 = 44$
Moyudan - Nanggulan - Kaligesing - Purworejo	Loano - Purworejo $f(n) = 12 + 37 = 49$
	Bagelen - Purworejo $f(n) = 19 + 39 = 58$
	Purwodadi - Purworejo $f(n) = 17 + 41 = 58$
Godean - Moyudan - Nanggulan - Kaligesing - Purworejo	Jogja - Godean - Moyudan - Nanggulan - Kaligesing - Purworejo $f(n) = 7 + 9 = 16$
	Samigaluh - Moyudan - Nanggulan - Kaligesing - Purworejo $f(n) = 18 + 26 = 44$

	Loano - Purworejo $f(n) = 12 + 37 = 49$
	Bagelen - Purworejo $f(n) = 19 + 39 = 58$
	Purwodadi - Purworejo $f(n) = 17 + 41 = 58$
Jogja - Godean - Moyudan - Nanggulan - Kaligesing - Purworejo	Solusi ditemukan

Berikut adalah visualisasi dari simpul-simpul yang telah diekspan dan bangkitkan.



Gambar 4. Visualisasi simpul eksplan dan simpul hidup

Dari gambar 4 terlihat jelas bahwa solusi untuk menentukan rute terpendek dari jalur alternatif Purworejo-Jogja telah didapatkan, dengan jalur sebagai berikut.

Purworejo - Kaligesing - Nanggulan - Moyudan - Godean - Jogja

Jalur yang menjadi solusi rute terpendek yang dihasilkan pada tabel 2 dan divisualisasikan pada gambar 4 merupakan jalur dengan *cost* (jarak) paling minimal diantara jalur yang lain dengan hasil perhitungan sebagai berikut.

Tabel 3. Hasil perhitungan jarak rute terpendek

Purworejo - Kaligesing	16 km
Kaligesing - Nanggulan	20 km
Nanggulan - Moyudan	12 km
Moyudan - Godean	7 km
Godean - Jogja	10 km
Total	65 km

Hasil perhitungan jarak dari rute terpendek tersebut merupakan hasil yang paling minimal di antara empat jalur alternatif lain. Jadi, terbukti bahwa algoritma A* dapat dijadikan solusi untuk mencari rute terpendek.

C. Pertimbangan dalam Menentukan Rute yang Akan Dipilih sebagai Jalur Alternatif Purworejo-Jogja

Rute terpendek dapat digunakan sebagai pilihan jalur tercepat saat melakukan perjalanan untuk mempersingkat waktu yang dihabiskan di jalan. Namun, perlu beberapa pertimbangan saat memilih jalur alternatif sebagai pilihan. Pada jalur alternatif Purworejo-Jogja yang telah penulis pilih, Berikut pertimbangan yang harus diperhatikan saat menentukan rute perjalanan, khususnya dari Purworejo menuju Jogja.

1. Perhatikan medan jalan yang akan dipilih. Pada jalur alternatif yang melalui Kaligesing, yang mana jalur tersebut adalah rute terpendek, jalur tersebut memiliki jalan yang menanjak curam sehingga perlu diperhatikan kondisi kendaraan yang akan dipakai untuk melalui jalur tersebut.
2. Pertimbangkan waktu saat melakukan perjalanan. Pada jalur alternative yang memiliki rute tercepat, mungkin waktu yang akan ditempuh juga cukup singkat dibandingkan yang lain, tetapi perlu mempertimbangkan hal lain, seperti medan jalan yang akan dilalui, karena itu juga akan mempengaruhi waktu perjalanan jika terdapat hambatan di jalan.
3. Pertimbangkan suasana pemandangan sekitar. Tentu jika melakukan perjalanan kita ingin menikmati perjalanan yang kita lalui, melewati jalur via Glagah (Jalan Daendels) mungkin akan memberi suasana baru saat di jalan. Pemandangan di sekitar jalur Kaligesing dominan pohon-pohon hijau yang akan membuat suasana menjadi sejuk, namun jalan berkelok. Jalur Bantul terdapat sawah di kanan kiri jalan sehingga dapat untuk menyegarkan mata Ketika melakukan perjalanan.

Hal-hal tersebut di atas merupakan pertimbangan saat menentukan jalur alternatif yang akan dilalui berdasarkan pengalaman penulis saat melakukan perjalanan ke Jogja dengan rute yang berbeda-beda.

IV. KESIMPULAN

Algoritma A* dapat digunakan untuk mencari rute terpendek dari beberapa jalur alternatif dari Purworejo menuju Jogja secara efisien. Hal tersebut terbukti dari proses penentuan rute terpendek tersebut yang menggunakan algoritma A* sesuai dengan informasi yang ada pada *google map*. Namun, rute terpendek tidak menjamin menjadi pilihan jalur alternatif dengan waktu yang singkat karena perlu mempertimbangkan segala sesuatu yang ada pada jalur alternatif tersebut, yang mana hambatan dan segala kondisi pada jalur alternatif tersebut tidak terdeteksi pula pada *google map* sehingga jika ingin melakukan perjalanan jangan hanya mempertimbangkan jarak saja.

VIDEO LINK AT YOUTUBE

<https://youtu.be/RPNTImRcqdw>

UCAPAN TERIMA KASIH

Pertama-tama, saya ucapkan syukur kepada Allah SWT karena berkat rahmat dan karunia-Nya, saya dapat menyelesaikan makalah dengan judul "*Penerapan Algoritma A* Menggunakan Graf untuk Menentukan Rute Terpendek dari Jalur Alternatif Purworejo-Jogja*" dengan sebaik-baiknya. Kedua, saya ucapkan terima kasih kepada Dr. Masayu Leylia Khodra selaku dosen mata kuliah strategi algoritma yang telah membimbing saya hingga berakhirnya semester empat ini. Selanjutnya, besar rasa terima kasih saya kepada kedua orang tua dan adik saya yang selalu mendoakan dan menjadi *support system* buat saya, khususnya bapak saya yang telah membantu saya dalam menentukan jalur-jalur alternatif dari Purworejo menuju Jogja. Tak lupa saya ucapkan terima kasih kepada teman-teman yang telah banyak membantu saya dalam mata kuliah ini.

REFERENSI

- [1] <https://fti.uajy.ac.id/sentika/publikasi/makalah/2018/9.pdf> diakses pada tanggal 3 Mei 2020.
- [2] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/A-Star-Best-FS-dan-UCS-\(2018\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/A-Star-Best-FS-dan-UCS-(2018).pdf) diakses pada 3 Mei 2020.
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Makalah/Makalah-Stima-2019-030.pdf> diakses pada tanggal 3 Mei 2020
- [4] <https://media.neliti.com/media/publications/103568-ID-none.pdf> diakses pada tanggal 3 Mei 2020.
- [5] <https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf> diakses pada tanggal 3 Mei 2020.
- [6] <https://www.google.co.id/maps/> diakses pada 2 Mei 2020

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Purworejo, 4 Mei 2020



Ferdina Wiranti Afifah
13518046