

Penggunaan Algoritma A* dalam Penentuan Jalur Mudik Terpendek dari Bandung ke Tasikmalaya

Azmi Muhammad Syazwana 13519151
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): azmisyazwana@gmail.com

Abstract—Setiap tahun di negara yang mayoritas penduduknya memeluk agama Islam pasti melakukan tradisi mudik. Saat mudik diperlukan jalur terpendek untuk mempercepat sampai ke tujuan menyambut hangatnya keluarga tercinta. Oleh karena itu, algoritma A* dapat dimanfaatkan untuk mencari jalur terpendek karena algoritma ini merupakan algoritma optimasi yang dapat mencari jalur yang paling optimal sehingga diperoleh jalur terpendek yang membuat mobilisasi mudik menjadi jauh lebih efektif, efisien, dan cepat.

Keywords—algoritma A*, graf, jalur terpendek, optimal

I. PENDAHULUAN

Pada era industri 4.0 ini, kita disajikan dengan berbagai macam teknologi yang sangat canggih, mulai dari *gadget* hingga alat transportasi. Kecanggihan tersebut terjadi karena manusia selalu ingin mencari suatu inovasi yang lebih efektif, lebih efisien, dan lebih bermanfaat. Hal itu lah yang membuat teknologi dari hari ke hari semakin berkembang pesat. Salah satu teknologi yang berkembang pesat, yaitu teknologi informasi.

Perkembangan yang sangat pesat dari teknologi informasi memengaruhi teknologi lainnya secara signifikan. Salah satu yang terpengaruh oleh perkembangan teknologi informasi pastinya, yaitu teknologi transportasi. Perkembangan transportasi ini sangat dipengaruhi teknologi informasi dan telekomunikasi serta pemanfaatan teknologi informasi yang secara langsung maupun tidak langsung memengaruhi penyelenggaraan transportasi.

Semakin pesatnya perkembangan teknologi transportasi, maka manusia akan semakin cepat melakukan mobilisasi atau perpindahan dari satu tempat ke tempat lainnya. Oleh karena itu, secara tidak langsung manusia membutuhkan waktu yang paling efektif dan efisien dalam melakukan segala aktivitas salah satunya dalam bidang mobilitas atau perpindahan.

Mobilisasi yang cepat, efektif, dan efisien akan sangat penting dan berguna karena waktu yang dibutuhkan untuk mobilisasi akan sangat minim sehingga waktu yang tersisa akan dapat digunakan untuk melakukan aktivitas lainnya. Selain faktor teknologi, untuk mendapatkan mobilisasi yang semakin cepat, efektif, dan efisien maka diperlukan jalur atau jarak terpendek yang dapat dilaluinya. Jika tidak begitu, mobilisasi yang dilakukan akan kurang optimal.

Salah satu contoh mobilisasi manusia terutama di Indonesia, yaitu budaya mudik saat hari raya Idul Fitri akan tiba. Saat mudik tiba, hampir setiap orang akan melakukan perjalanan menggunakan berbagai alat transportasi untuk bertemu keluarga besar di kampung halamannya. Ada berbagai macam transportasi yang digunakan, seperti mobil, motor, bus, kereta api, pesawat, dan kapal laut. Untuk pengguna jalur darat memerlukan rute atau jalur terpendek agar mobilisasi yang dilakukan sangat optimal dengan artian mobilisasi yang dilakukan adalah mobilisasi yang paling cepat, paling efektif dan paling efisien.

Untuk memperjelas hal tersebut, penulis mengambil contoh jalur mudik penulis, yaitu dari Bandung ke Tasikmalaya. Dalam makalah ini, permasalahan tersebut termasuk ke dalam *route/path planning*. Permasalahan tersebut akan dijelaskan dalam makalah ini dengan menggunakan algoritma A* dalam menentukan jalur terpendek sehingga mobilisasi yang dilakukan adalah mobilisasi yang paling optimal.

II. LANDASAN TEORI

A. Graf

Graf adalah suatu kumpulan simpul tak kosong yang dihubungkan dengan simpul yang lainnya melalui suatu sisi atau busur. Sebuah graf terdiri dari dua himpunan, yaitu himpunan V (*vertices*) dan himpunan E (*edges*). Himpunan V, yaitu himpunan dari simpul yang tidak kosong, sedangkan himpunan E adalah himpunan sisi atau busur yang menghubungkan sepasang simpul. Dengan kata lain, v atau *vertices* adalah simpul dan e atau *edges* adalah sisi atau busur. Penulisan graf dilakukan seperti berikut, $G = (V, E)$.

Jenis-jenis graf sangat banyak dan dibedakan dengan berbagai kriteria sebagai berikut:

1. Berdasarkan ada tidaknya sisi ganda

Berdasarkan hal ini suatu graf dikelompokkan menjadi dua, yaitu graf sederhana (*simple graph*) dan graf tak-sederhana (*unsimple-graph*). Graf sederhana yaitu, graf yang tidak mempunyai sisi ganda, sedangkan graf tak-sederhana adalah gelah yang memiliki sisi ganda.

2. Berdasarkan orientasi arah pada busur atau sisi

Berdasarkan hal ini, suatu graf dibedakan menjadi dua jenis, yaitu graf berarah (*directed graph* atau *digraph*) dan graf tak-berarah (*undirected graph*). Graf berarah yaitu graf yang setiap sisinya memiliki arah yang menuju suatu simpul, sedangkan graf tak-berarah adalah graf yang sisinya atau busurnya tidak memiliki arah.

Graf memiliki berbagai terminologi yang sangat banyak seperti berikut, ketetanggaan, bersisian, simpul terpercil, graf kosong, derajat, lintasan, siklus atau sirkuit, keterhubungan, upagraf dan komplemen upagraf, upagraf merentang, *cut-set*, dan graf berbobot. Namun, tidak semua terminologi akan dipakai dalam menyelesaikan permasalahan dalam makalah ini. Hanya ada lima terminologi yang akan dipakai, yaitu ketetanggaan, bersisian, lintasan, keterhubungan, dan graf berbobot.

1. Ketetanggaan (*Adjacent*)

Ketetanggaan terjadi jika dua buah simpul langsung terhubung. Jika terdapat sisi yang menghubungkan simpul u dan v , maka u bertetangga dengan v dan juga sebaliknya v bertetangga dengan u .

2. Bersisian (*Incidency*)

Bersisian artinya sebuah sisi atau busur akan bersisian dengan simpul yang langsung dihubungkannya. Misalnya, terdapat sisi e yang menghubungkan simpul u dan v , maka e bersisian dengan simpul u atau e bersisian dengan simpul v .

3. Lintasan (*Path*)

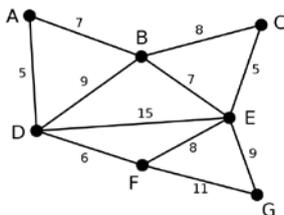
Lintasan adalah jalur sisi atau busur yang dilalui dari simpul awal ke simpul tujuan di dalam graf. Panjang lintasan, yaitu jumlah sisi dalam lintasan tersebut.

4. Keterhubungan (*Connected*)

Keterhubungan terjadi jika terdapat lintasan antara dua buah simpul. Graf G disebut graf terhubung jika setiap pasang simpul u dan v terdapat lintasan dari u ke v .

5. Graf Berbobot (*Weighted Graph*)

Graf berbobot adalah sebuah graf yang memiliki bobot atau harga di setiap sisi-sisinya.



Gambar 2.1 Contoh Graf Berbobot

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

B. Algoritma A*

Algoritma A* merupakan salah satu algoritma untuk penentuan rute (*route/path Planning*). Algoritma ini juga termasuk ke dalam algoritma *informed search* atau sudah diketahui informasinya seperti algoritma GBFS (*Greedy Best First Search*). Berbeda dengan algoritma pencarian lainnya yang termasuk ke dalam kategori *uninformed search* atau *blind search* seperti, BFS (*Breadth First Search*), DFS (*Depth First Search*), DLS (*Depth Limited Search*), IDS (*Iterative Deepening Search*), dan UCS (*Uniform Cost Search*).

Maksud dari *informed search*, yaitu algoritma ini melakukan pencarian dengan informasi tambahan berbasis heuristik. Algoritma A* merupakan perbaikan dari algoritma BFS (*Breadth First Search*) dengan memodifikasi fungsi heuristiknya. Algoritma ini juga merupakan algoritma optimasi sehingga algoritma ini sangat populer dalam ilmu komputer dalam masalah pencarian rute pada suatu graf.

Optimasi algoritma A* akan meminimumkan total biaya lintasan atau akan mengoptimalkan ongkos dengan tetap memperhatikan batasan-batasan dari suatu permasalahannya. Algoritma ini akan memberikan solusi yang terbaik dan waktu yang optimal jika terdapat pada kondisi yang tepat.

Algoritma A* merupakan gabungan dari dua dasar kerja algoritma lainnya, yaitu algoritma UCS (*Uniform Cost Search*) dan algoritma GBFS (*Greedy Best First Search*). Untuk mencari biaya yang paling rendah menggunakan algoritma UCS (*Uniform Cost Search*) dimulai dari titik awal hingga titik akhir. Untuk memberikan estimasi biaya dari titik awal hingga titik akhir menggunakan algoritma GBFS (*Greedy Best First Search*) dengan menggunakan fungsi heuristik. Pencarian jalur atau rute tercepat merupakan salah satu dari implementasi algoritma A*.

Pada algoritma A* juga memanfaatkan prinsip yang dianut algoritma BFS (*Breadth First Search*) untuk membangkitkan simpul. Algoritma ini juga menyerupai algoritma UCS (*Uniform Cost Search*) dan algoritma GBFS (*Greedy Best First Search*) dalam membentuk pohon pencariannya.

Algoritma A* juga merupakan algoritma pencarian yang menganalisis inputnya lalu mengevaluasi jalur yang mungkin dilalui sehingga menghasilkan solusi. Algoritma ini juga digunakan secara luas dalam ilmu komputer pada graph traversal dan penemuan jalur yang dapat dilewati secara efektif dan efisien.

Pada algoritma A*, simpul-simpul (nodes) dievaluasi dengan fungsi evaluasi f yang dinyatakan sebagai berikut:

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

Keterangan:

$f(n)$: estimasi biaya termurah melalui jalur solusi yang melewati n

$g(n)$: nilai biaya atau ongkos untuk mencapai simpul n dari simpul akar

$h(n)$: nilai heuristic (nilai ongkos atau biaya dari simpul n menuju simpul tujuan)

Nilai $g(n)$ dan $h(n)$ pada fungsi evaluasi ini masing-masing menyerupai fungsi objektif pada algoritma lain, yaitu nilai $g(n)$ menyerupai fungsi objektif pada algoritma UCS (*Uniform Cost Search*), sedangkan nilai $h(n)$ menyerupai fungsi objektif pada algoritma GBFS (*Greedy Best First Search*)

Nilai heuristic dapat diartikan sebagai sebuah fungsi yang memberikan nilai berupa biaya estimasi dari suatu solusi. Dengan menggunakan teknik pencarian heuristic dapat menentukan solusi pencarian yang kemungkinan suksesnya paling besar karena teknik pencarian ini menggunakan strategi untuk melakukan proses pencarian secara selektif, tetapi mengorbankan kelengkapannya. Diperlukan suatu fungsi heuristic untuk menerapkan pencarian heuristic.

Nilai heuristic juga dapat menentukan kualitas solusi pada simpul tersebut dan kerumitan penyelesaian masalah pada suatu simpul. Nilai heuristic tidak boleh ditentukan secara sembarangan. Ada ketentuan atau aturan yang harus diikuti untuk menentukan nilai heuristic. Jadi ada aturan apakah nilai heuristic tersebut dapat diterima (*admissible*) atau tidak dapat diterima (*not admissible*). Syarat nilai heuristic dapat dikatakan diterima (*admissible*), yaitu nilai heuristic tidak boleh lebih besar dari nilai sebenarnya. Contoh, jika menghitung jarak dari suatu kota ke kota lain maka nilai heuristicnya tidak boleh lebih besar dari nilai jarak aslinya.

Syarat atau aturan nilai heuristic dapat diterima (*admissible*) atau tidak dapat diterima (*not admissible*) dinyatakan sebagai berikut:

$$h(n) \leq h^*(n)$$

Keterangan:

$h(n)$: nilai heuristic biaya dari simpul n untuk mencapai simpul tujuan

$h^*(n)$: nilai biaya sebenarnya dari simpul n untuk mencapai simpul tujuan

Algoritma A* ini pasti akan mempunyai penyelesaiannya atau simpul tujuannya pasti akan tercapai. Hal ini berarti algoritma A* bersifat komplet. Algoritma A* memiliki kompleksitas waktu $O(b^m)$ untuk kompleksitas waktu dan ruang dengan banyaknya suatu tingkat pohon pencarian merupakan nilai m serta banyaknya simpul anak yang mungkin diekspan adalah nilai b .

Algoritma A* memiliki beberapa properti sebagai berikut:

1. Bersifat komplet
2. Kompleksitas waktunya $O(b^m)$
3. Kompleksitas ruangnya $O(b^m)$
4. Optimal

C. Jalur Terpendek

Jalur terpendek atau biasa disebut dengan *shortest path* adalah lintasan terpendek yang diperlukan dari tempat awal untuk mencapai ke suatu tempat tertentu. Jalur terpendek (*shortest path*) ini memiliki persoalan, yaitu mencari atau menentukan jalur terpendek antara dua buah simpul atau lebih yang saling berhubungan. Masalah ini merupakan salah satu masalah yang memerlukan optimasi serta permasalahan ini biasanya direpresentasikan dalam bentuk graf.

D. Mudik



Gambar 2.2 Ilustrasi Mudik

Sumber:

<https://ayoyogya.com/read/2020/04/23/39201/mencermati-larangan-mudik-saat-pandemi>

Kata mudik berasal dari kata “udik” yang berarti selatan atau hulu. Jadi, pada zaman dahulu masih banyak wilayah yang bernama akhir udik atau ilir (utara atau hilir) sebelum di Jakarta terjadi urbanisasi. Pada saat itu, suplai hasil bumi dari kota Jakarta diambil dari wilayah di luar kota sehingga para pedagang banyak yang membawa dagangannya melalui sungai. Oleh karena itu, munculah istilah milir-mudik yang berarti bolak-balik. Dari situlah kemudian muncul kata mudik.

Mudik merupakan salah satu budaya di Indonesia yang sudah ada sejak lama. Mudik adalah kegiatan mobilisasi atau perpindahan yang dilakukan oleh pekerja migran dari tempat kerja atau tempat tinggalnya ke kampung halamannya. Pada saat itu lah setiap orang memiliki kesempatan untuk berkumpul dengan semua anggota keluarga besar untuk menghabiskan waktu dan menjalin silaturahmi.

Transportasi yang digunakan pun beragam mulai dari kendaraan darat, laut, serta udara, seperti mobil, motor, bus,

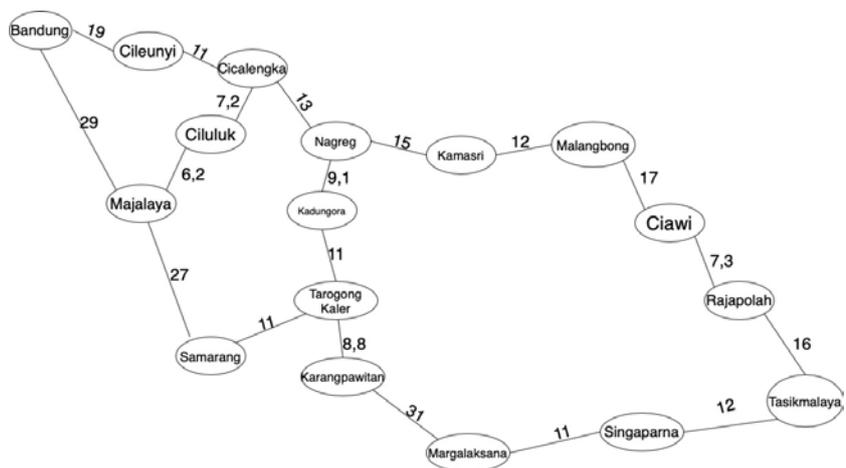
kereta api, kapal laut, dan pesawat. Tradisi atau budaya mudik ini biasanya terjadi di negara yang mayoritas penduduknya memeluk agama islam, seperti Indonesia, Malaysia, dan Bangladesh.

III. ANALISIS DAN PEMBAHASAN

Bagian ini akan berfokus pada proses implementasi algoritma A* dalam menentukan jalur mudik terpendek dari Bandung ke Tasikmalaya.

A. Graf Jalur dari Bandung ke Tasikmalaya

Untuk menerapkan algoritma A* dalam menentukan rute atau jalur terpendek dari Bandung ke Tasikmalaya diperlukan visualisasi peta dalam bentuk graf. Berikut adalah visualisasi peta jalur dari Bandung ke Tasik dalam bentuk graf yang diambil dari *google maps*.



Gambar 3.2 Penyederhanaan Visualisasi Graf Jalur Bandung-Tasikmalaya

Visualisasi graf pada gambar 3.2 telah disederhanakan agar menjadi lebih simpel sehingga lebih mudah dimengerti serta telah diberi bobot pada setiap simpulnya, yaitu jarak dari simpul ke simpul lainnya dalam satuan kilometer (km). Pada graf gambar 3.2 dapat dilihat berbagai jalur dan daerah yang dapat dilalui jika ingin pergi ke Tasikmalaya dari Bandung.

Dengan menggunakan graf pada gambar 3.2, penulis dapat lebih mudah mencari jalur terpendeknya dan akan lebih mudah pula dalam mengimplementasikan algoritma A*.

B. Penerapan Algoritma A* untuk Menentukan Jalur Terpendek dari Bandung ke Tasikmalaya

Untuk menyelesaikan persoalan ini penulis menggunakan algoritma A*. Dalam menggunakan algoritma A* perlu terlebih dahulu menentukan rumus fungsi evaluasinya. Rumus fungsi evaluasi yang digunakan dalam permasalahan ini dinyatakan sebagai berikut.

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

Keterangan:

$f(n)$: estimasi biaya termurah melalui jalur solusi yang melewati n

$g(n)$: jarak tempuh dari simpul awal ke simpul n.

$h(n)$: nilai heuristik, yaitu jarak yang ditempuh dari simpul n ke simpul tujuan yang diambil dengan garis lurus (jarak secara garis lurus).

Data nilai heuristik atau jarak secara garis lurus dari tiap simpul ditentukan dari fitur *google maps* yaitu *measure distance*. Berikut merupakan data nilai heuristik yang dimiliki oleh setiap simpul.

Gambar 3.1 Visualisasi Graf Jalur Bandung-Tasikmalaya

Sumber:
Google Maps

Pada gambar 3.1 terlihat titik-titik merah yang divisualisasikan sebagai simpul dari graf yang merupakan daerah jalur yang dapat dilalui dari Bandung untuk menuju ke Tasikmalaya.

Selanjutnya, penulis melakukan penyederhanaan graf yang terdapat pada gambar 3.1 menjadi sebagai berikut.

Tabel 3.1 Nilai h(n) tiap simpul

Simpul	h(n) (km)
Bandung	80
Cileunyi	67,9
Majalaya	60,1
Cicalengka	60,15
Nagreg	48
Ciluluk	58,8
Kamasari	39,35
Kadungora	46
Tarogong Kaler	40,65
Samarang	44,25
Malangbong	34,35
Karangpawitan	37,8
Ciawi	18
Margalaksana	19
Rajapolah	12,5
Singaparna	10,5

Langkah-langkah yang dilakukan untuk menyelesaikan permasalahan ini menggunakan algoritma A* adalah sebagai berikut:

1. Simpul awal yang dibangkitkan, yaitu simpul akar. Simpul akar pada masalah ini, yaitu Bandung.
2. Bangkitkan simpul yang akan menjadi simpul hidup, yaitu simpul yang berhubungan langsung dengan simpul akar.
3. Simpul hidup dievaluasi menggunakan nilai $f(n)$. Simpul hidup yang memiliki nilai $f(n)$ terkecil kemudian di ekspan menjadi simpul ekspan. Langkah ini dilakukan hingga mencapai solusi
4. Matikan semua simpul hidup saat solusi telah ditemukan

Berikut adalah tabel simpul ekspan, simpul hidup, dan nilai $f(n)$ yang merupakan implementasi algoritma A* dalam menentukan jalur terpendek dari Bandung ke Tasikmalaya

Tabel 3.2 Implementasi algoritma A*

Simpul Ekspan	Simpul Hidup	$f(n) = g(n) + h(n)$
Bandung	Cileunyi-Bandung	$f(n) = 19 + 67,9 = 86,9$
	Majalaya-Bandung	$f(n) = 29 + 60,1 =$

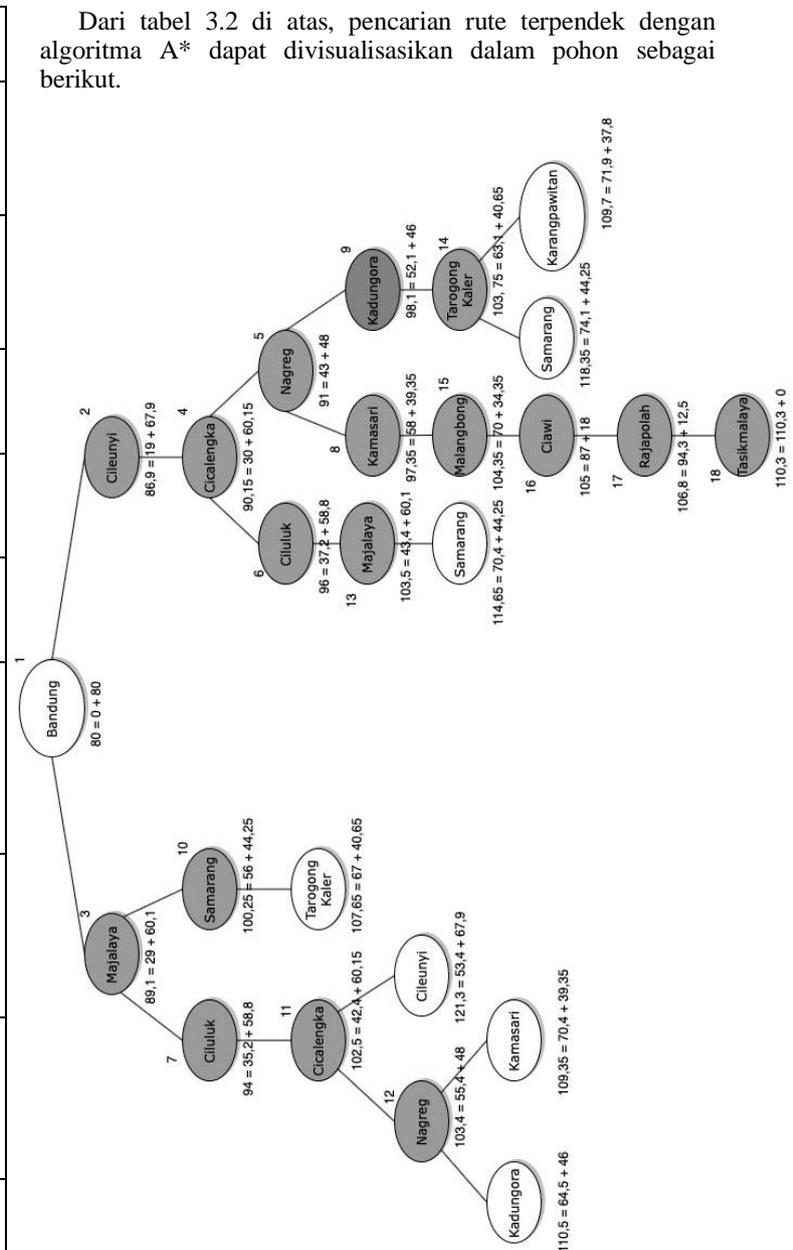
		89,1
Cileunyi-Bandung	Cicalengka - Cileunyi-Bandung	$f(n) = 30 + 60,15 = 90,15$
	Majalaya-Bandung	$f(n) = 29 + 60,1 = 89,1$
Majalaya-Bandung	Ciluluk-Majalaya-Bandung	$f(n) = 35,2 + 58,8 = 94$
	Samarang-Majalaya-Bandung	$f(n) = 56 + 44,25 = 100,25$
	Cicalengka - Cileunyi-Bandung	$f(n) = 30 + 60,15 = 90,15$
Cicalengka - Cileunyi-Bandung	Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 43 + 48 = 91$
	Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 37,2 + 58,8 = 96$
	Ciluluk-Majalaya-Bandung	$f(n) = 35,2 + 58,8 = 94$
	Samarang-Majalaya-Bandung	$f(n) = 56 + 44,25 = 100,25$
Nagreg-Cicalengka-Cileuyi-Bandung	Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 58 + 39,35 = 97,35$
	Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 52,1 + 46 = 98,1$
	Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 37,2 + 58,8 = 96$
	Ciluluk-Majalaya-Bandung	$f(n) = 35,2 + 58,8 = 94$
	Samarang-Majalaya-Bandung	$f(n) = 56 + 44,25 = 100,25$
Ciluluk-Majalaya-Bandung	Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 42,4 + 60,15 = 102,5$
	Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 58 + 39,35 = 97,35$
	Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 52,1 + 46 = 98,1$
	Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 37,2 + 58,8 = 96$
	Samarang-Majalaya-Bandung	$f(n) = 56 + 44,25 = 100,25$
Ciluluk-	Majalaya-Ciluluk-	$f(n) = 43,4 + 60,1$

Cicalengka-Cileuyi-Bandung	Cicalengka-Cileuyi-Bandung	$= 103,5$
	Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 42,4 + 60,15 = 102,5$
	Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 58 + 39,35 = 97,35$
	Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 52,1 + 46 = 98,1$
	Samarang-Majalaya-Bandung	$f(n) = 56 + 44,25 = 100,25$
Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 70 + 34,35 = 104,35$
	Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 43,4 + 60,1 = 103,5$
	Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 42,4 + 60,15 = 102,5$
	Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 52,1 + 46 = 98,1$
	Samarang-Majalaya-Bandung	$f(n) = 56 + 44,25 = 100,25$
	Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung
Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung		$f(n) = 70 + 34,35 = 104,35$
Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung		$f(n) = 43,4 + 60,1 = 103,5$
Cicalengka-Ciluluk-Majalaya-Bandung		$f(n) = 42,4 + 60,15 = 102,5$
Samarang-Majalaya-Bandung		$f(n) = 56 + 44,25 = 100,25$
Samarang-Majalaya-Bandung		Tarogong Kaler-Samarang-Majalaya-Bandung
	Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 63,1 + 40,65 = 103,75$

	Bandung	
	Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 70 + 34,35 = 104,35$
	Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 43,4 + 60,1 = 103,5$
	Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 42,4 + 60,15 = 102,5$
Cicalengka-Ciluluk-Majalaya-Bandung	Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 55,4 + 48 = 103,4$
	Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 53,4 + 67,9 = 121,3$
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$
	Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 63,1 + 40,65 = 103,75$
	Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 70 + 34,35 = 104,35$
	Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 43,4 + 60,1 = 103,5$
	Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	Kadungora-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung
Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	Kamasari-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 39,35 = 109,35$
	Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 53,4 + 67,9 = 121,3$
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$
	Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 63,1 + 40,65 = 103,75$
	Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 70 + 34,35 = 104,35$

	Bandung				
	Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 43,4 + 60,1 = 103,5$			
Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	Samarang-Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 70,4 + 44,25 = 114,65$			
	Kadungora-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 64,5 + 46 = 110,5$			
	Kamasari-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 39,35 = 109,35$			
	Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 53,4 + 67,9 = 121,3$			
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$			
	Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 63,1 + 40,65 = 103,75$			
	Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 70 + 34,35 = 104,35$			
	Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	Samarang-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 74,1 + 44,25 = 118,35$		
Karangpawitan-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung		$f(n) = 71,9 + 37,8 = 109,7$			
Samarang-Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung		$f(n) = 70,4 + 44,25 = 114,65$			
Kadungora-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung		$f(n) = 64,5 + 46 = 110,5$			
Kamasari-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung		$f(n) = 70,4 + 39,35 = 109,35$			
Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung		$f(n) = 53,4 + 67,9 = 121,3$			
Tarogong Kaler-Samarang-Majalaya-Bandung		$f(n) = 67 + 40,65 = 107,65$			
Ciawi-Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung		Rajapolah-Ciawi-Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 94,3 + 12,5 = 106,8$		
	Samarang-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 74,1 + 44,25 = 118,35$			
	Karangpawitan-Tarogong Kaler-Kadungora-Nagreg-	$f(n) = 74,1 + 44,25 = 118,35$			
	Majalaya-Bandung				
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$			
	Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 70 + 34,35 = 104,35$			
	Ciawi-Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 87 + 18 = 105$			
	Samarang-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 74,1 + 44,25 = 118,35$			
	Karangpawitan-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 71,9 + 37,8 = 109,7$			
	Samarang-Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 70,4 + 44,25 = 114,65$			
	Kadungora-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 64,5 + 46 = 110,5$			
	Kamasari-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 39,35 = 109,35$			
	Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 53,4 + 67,9 = 121,3$			
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$			
	Ciawi-Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 94,3 + 12,5 = 106,8$			
	Samarang-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 74,1 + 44,25 = 118,35$			
	Karangpawitan-Tarogong Kaler-Kadungora-Nagreg-	$f(n) = 74,1 + 44,25 = 118,35$			

	Cicalengka-Cileuyi-Bandung	
	Samarang-Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 71,9 + 37,8 = 109,7$
	Kadungora-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 44,25 = 114,65$
	Kamasari-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 39,35 = 109,35$
	Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 53,4 + 67,9 = 121,3$
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$
Rajapolah-Ciawi-Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	Tasikmalaya-Rajapolah-Ciawi-Malangbong-Kamasari-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 110,3 + 0 = 110,3$
	Samarang-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 74,1 + 44,25 = 118,35$
	Karangpawitan-Tarogong Kaler-Kadungora-Nagreg-Cicalengka-Cileuyi-Bandung	$f(n) = 74,1 + 44,25 = 118,35$
	Samarang-Majalaya-Ciluluk-Cicalengka-Cileuyi-Bandung	$f(n) = 71,9 + 37,8 = 109,7$
	Kadungora-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 44,25 = 114,65$
	Kamasari-Nagreg-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 70,4 + 39,35 = 109,35$
	Cileunyi-Cicalengka-Ciluluk-Majalaya-Bandung	$f(n) = 53,4 + 67,9 = 121,3$
	Tarogong Kaler-Samarang-Majalaya-Bandung	$f(n) = 67 + 40,65 = 107,65$



Gambar 3.2 Pohon pencarian rute terpendek menggunakan algoritma A*

Dari gambar 3.2 dapat dilihat bahwa solusi paling optimalnya atau jalur terpendek dari Bandung ke Tasikmalaya, yaitu Bandung – Cileunyi – Cicalengka – Nagreg – Kamasari – Malangbong – Ciawi – Rajapolah – Tasikmalaya. Total jarak yang ditempuhnya yaitu 110,3 km.

Identify applicable sponsor/s here. If no sponsors, delete this text box (sponsors).

IV. KESIMPULAN

Materi mata kuliah Strategi Algoritma sangat bermanfaat baik bagi pendidikan maupun bagi kehidupan sehari-hari karena terbukti setiap materi dapat diimplementasikan untuk menyelesaikan persoalan-persoalan yang ada di dunia. Salah satu materi yang sangat bermanfaatnya, yaitu algoritma A*

Algoritma A* sangat dapat diimplementasikan pada persoalan kehidupan sehari-hari, salah satunya dalam menentukan jalur terpendek (*shortest path*). Algoritma ini dapat mencari jalur yang paling optimal karena algoritma ini merupakan algoritma optimasi. Oleh karena itu, algoritma ini diharapkan dapat membantu menyelesaikan berbagai macam permasalahan yang memerlukan optimasi, seperti membuat mobilisasi yang dilakukan menjadi jauh lebih efektif, efisien, dan cepat.

PRANALA VIDEO YOUTUBE

bit.ly/VideoMakalahStima

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya penulis bisa menyelesaikan tugas makalah ini yang berjudul "Penggunaan Algoritma A* dalam Penentuan Jalur Mudik Terpendek dari Bandung ke Tasikmalaya". Selain itu, tidak lupa penulis ucapkan terima kasih kepada kedua orang tua, keluarga, dan teman-teman yang selalu memberikan dukungan.

Penulis juga mengucapkan terima kasih kepada seluruh dosen mata kuliah IF2211 atau Strategi Algoritma ini terutama Pak Prof.Ir. Dwi Hendratmo Widyantoro, M.Sc.,Ph.D., selaku pembimbing dan dosen K3. Karena tanpa beliau, penulis tidak akan mengerti tentang materi yang disampaikan pada makalah ini.

REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf> diakses pada tanggal 10 Mei 2021 pukul 09.00 WIB
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian1-2021.pdf> diakses pada tanggal 10 Mei 2021 pukul 09.00 WIB
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf> diakses pada tanggal 10 Mei 2021 pukul 09.00 WIB
- [4] <http://dephub.go.id/post/read/menhub-pemanfaatan-teknologi-informasi-dan-telekomunikasi-mempengaruhi-penyelenggaraan-transportasi> diakses pada tanggal 10 Mei 2021 pukul 12.00 WIB
- [5] <https://fti.uajy.ac.id/sentika/publikasi/makalah/2018/9.pdf> diakses pada tanggal 10 Mei 2021 pukul 12.00 WIB

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.

Bandung, 11 Mei 2021



Azmi Muhammad Syazwana
13519151