

*Penentuan Rute terpendek dari Jalur Alternatif Padang - Pekanbaru dengan Menggunakan Graf dengan Algoritma A**

Ilham Pratama - 13520041

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13520041@std.stei.itb.ac.id

Abstract—Algoritma A* merupakan suatu algoritma yang terbentuk dari gabungan algoritma Uniform Cost dan Algoritma greedy Best First Search. Algoritma Uniform Cost merupakan algoritma yang digunakan untuk menghitung biaya yang telah dikeluarkan dari suatu keadaan awal sampai ke suatu node, sedangkan algoritma Greedy Best First Search digunakan untuk menghitung biaya tambahan yang harus dikeluarkan dari suatu node hingga mencapai keadaan akhir dengan menggunakan suatu fungsi heuristic. Algoritma A* digunakan untuk menentukan rute terpendek. Algoritma A* menghasilkan rute terpendek yang efisien berdasarkan jarak yang ditempuh dan straight-line distance yang merupakan jarak lurus dari suatu titik kota menuju node tujuannya. Penentuan rute terpendek dari jalur alternatif Padang-Pekanbaru 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 alternatif pada rute perjalanan Padang-Pekanbaru.

Keywords—algoritma A*; Uniform Cost; greedy Best First Search; rute terpendek; jalur alternatif; Padang; Pekanbaru.

I. PENDAHULUAN

Pada era industri 4.0, kita disajikan berbagai macam teknologi yang sangat canggih mulai dari gadget hingga alat transportasi yang memiliki mobilitas yang tinggi. Kemajuan tersebut terjadi karena manusia ingin mencari suatu inovasi baru yang lebih efektif, efisien dan lebih bermanfaat. Hal tersebut membuat teknologi dari hari ke hari menjadi semakin berkembang pesat. Salah satu teknologi yang berkembang pesat adalah teknologi informasi.

Perkembangan teknologi informasi yang begitu pesat mengakibatkan teknologi transportasi juga mengalami perkembangan. Hal tersebut terjadi karena semakin banyak informasi yang didapatkan terkait dengan rute perjalanan dari suatu transportasi, baik berupa medan area dari rute tersebut maupun kepadatan lalu lintas yang terjadi pada rute tersebut. Dengan berkembangnya teknologi transportasi, manusia akan semakin cepat melakukan mobilisasi. 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.

Padang-Pekanbaru merupakan suatu rute perjalanan yang terdapat di Pulau Sumatra. Kota Padang adalah kota terbesar di pantai barat Pulau Sumatra dan ibu kota provinsi Sumatra Barat, Indonesia. Kota ini merupakan pintu gerbang barat Indonesia dari Samudra Hindia. Kota Padang sendiri memiliki masyarakat yang senang merantau. Banyak masyarakat dari Kota Padang yang merantau ke berbagai daerah di seluruh Indonesia, salah satu daerah tersebut merupakan Kota Pekanbaru. Kota Pekanbaru merupakan ibu kota dan kota terbesar di provinsi Riau, Indonesia. Kota ini merupakan salah satu sentra ekonomi terbesar di pulau Sumatra dan termasuk kota dengan tingkat pertumbuhan, migrasi, dan urbanisasi yang tinggi.

Dengan banyaknya masyarakat Kota Padang yang merantau ke kota Pekanbaru, maka tak heran bahwa banyak kendaraan transportasi yang berlalu-lintas pada rute Padang-Pekanbaru ini. Baik itu kendaraan yang menuju ke Kota Padang dari Kota Pekanbaru, maupun dari Kota Padang menuju kota Pekanbaru. Ada banyak jalur alternatif yang dapat dilalui untuk menuju ke kedua kota tersebut. Namun ketika akan melakukan perjalanan dari suatu tempat ke tempat lain, seseorang pasti sangat mempertimbangkan berbagai hal, seperti efisiensi biaya dan waktu. Oleh karena itu, diperlukan jalur tercepat yang bisa ditempuh untuk dari Kota Padang ke Kota Bukittinggi. Algoritma pencarian rute terpendek atau lebih dikenal sebagai shortest-path dipakai dalam menentukan rute dalam sebuah graf.

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 jalur 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. Salah satu metode pencarian jalur terpendek yang termasuk dalam pencarian heuristic adalah Algoritma A*. Untuk mencapai tujuan dengan jarak tempuh terdekat, Algoritma A* memiliki suatu nilai heuristik yang digunakan sebagai dasar pertimbangan dimana estimasi nilai/biaya terkecil yang akan menentukan jarak tempuh terdekat.

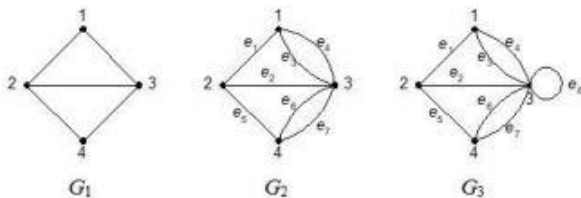
II. LANDASAN TEORI

A. Graf

Graf merupakan 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,

$$G = (V, E)$$

Contoh graf :



Gambar 1. Contoh graf
(sumber :

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

Pada gambar 1, graf G1 terdiri dari

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

$$E = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)\}$$

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

1. Berdasarkan ada tidaknya sisi ganda

Ada 2 macam, yaitu

- Graf sederhana,
Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda
- Graf tak-sederhana
Graf tak-sederhana adalah graf yang mengandung sisi gelang atau sisi ganda

2. Berdasarkan orientasi arah pada busur

Ada 2 macam, yaitu:

- Graf berarah
Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah
- Graf tak-berarah
Graf tak-berarah adalah graf yang sisinya tidak memiliki orientasi arah

3. Berdasarkan jumlah simpul

Ada 2 macam, yaitu

- Graf berhingga
Graf berhingga adalah graf yang jumlah simpulnya, n, berhingga.
- Graf tak-berhingga
Graf tak berhingga adalah Graf yang jumlah simpulnya, n, tidak berhingga banyaknya.

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 adalah : (Pawitri, dkk : 2007)

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

C. Tree

Tree merupakan struktur data yang mempunyai hubungan one to many. Hubungan one to many ini meliputi juga hubungan one to one atau one to zero, dapat dijelaskan bahwa satu parent bisa memiliki satu atau nol atau lebih dari satu child. Elemen dalam tree disebut dengan node, tree memiliki karakteristik sebagai berikut.

- Terdapat satu node yang unik, yang tidak memiliki predecessor. Node ini disebut root.
- Terdapat satu atau beberapa node yang tidak mempunyai successor. Node ini disebut leaf.
- Setiap node kecuali root, pasti memiliki satu predecessor yang unik.

D. 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

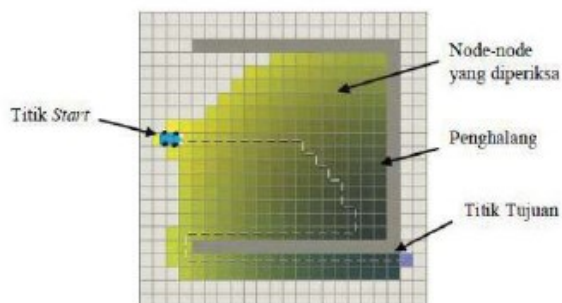
lainnya yang termasuk ke dalam kategori *uninformed search* atau *blind search* seperti, BFS (*Breadth First Search*), DFS (*Depth First Search*), DLS (*Depth Limited Search*), IDS (*Iterative Deepening Search*), dan UCS (*Uniform Cost Search*).

Maksud dari *informed search*, yaitu algoritma ini melakukan pencarian dengan informasi tambahan berbasis heuristic. 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.

Algoritma A* adalah algoritma yang seringkali digunakan dalam pencarian jalan dan traversal graf (Setiawan, 2010). Algoritma ini diciptakan oleh Peter Hart, Nils Nilsson, dan Bertram Raphael pada tahun 1968. A* menggunakan best first search dan bobot minimal yang diberikan dari simpul awal ke simpul tujuan. A* menggunakan jarak ditambah biaya (biasa dinyatakan $f(x)$) dari fungsi heuristik untuk menentukan urutan pencarian mencapai simpul mana pada sebuah pohon. Biaya jarak ditambah heuristik terdiri dari 2 buah fungsi. Pertama adalah fungsi biaya jarak, yang merupakan biaya dari simpul awal sampai simpul sekarang (biasa dinyatakan dengan $g(x)$). Kedua adalah, sebuah perkiraan heuristik dari jarak ke tujuan (biasa dinyatakan $h(x)$).

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 heuristic. Pencarian jalur atau rute tercepat merupakan salah satu dari implementasi algoritma A*.

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)



Gambar 2. Path Finding pada A*
(sumber :

<https://sinta.unud.ac.id/uploads/wisuda/1108605031-3-bab2.pdf>)

Starting point adalah sebuah terminologi posisi awal sebuah benda. A adalah simpul yang sedang dijalankan algoritma pencarian jalan terpendek. Simpul adalah petak-petak kecil sebagai representasi dari area pathfinding. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga. open list adalah tempat menyimpan data simpul yang mungkin diakses dari starting point maupun simpul yang sedang dijalankan. Closed list adalah tempat menyimpan data simpul sebelum A yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan. Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari starting point ke A, dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Simpul tujuan yaitu simpul yang dituju. Rintangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A.

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

Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (starting point) menuju simpul tujuan dengan memperhatikan harga (F) terkecil (Victor et al., 2005). A* memperhitungkan cost dari current state ke tujuan dengan fungsi heuristik, Fungsi yang digunakan dalam pencarian jalur terpendek dengan algoritma ini adalah :

$$f = g + h$$

dimana,

g = biaya pergerakan dari current node ke sebuah petak
h = heuristik

Algoritma ini juga mempertimbangkan cost yang telah ditempuh selama ini dari initial state ke current state. Jadi jika ada jalan yang telah ditempuh sudah terlalu panjang dan ada jalan lain yang cost-nya lebih kecil tetapi memberikan posisi yang sama dilihat dari goal, jalan yang lebih pendek yang akan dipilih.

E. Fungsi Heuristik

Pada pencarian heuristik, kata heuristik berasal dari sebuah kata kerja bahasa Yunani, heuriskein, yang berarti "Mencari atau Menemukan". Dalam dunia pemrograman, sebagian orang menggunakan kata heuristik sebagai lawan kata dari algoritma, dimana kata heuristik 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, kata heuristic diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi. Teknik pencarian heuristik (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

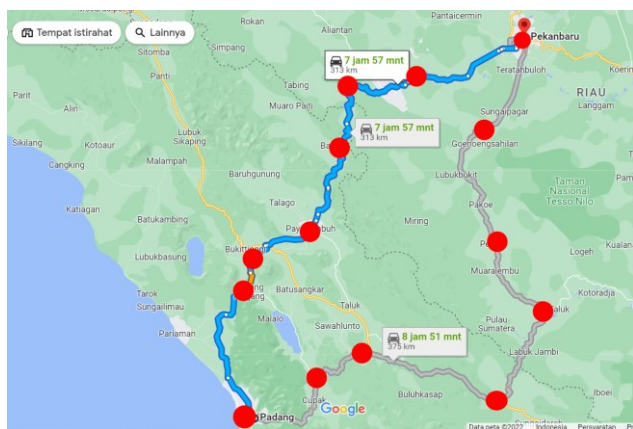
kelengkapan (completeness). Untuk menerapkan pencarian heuristik diperlukan suatu fungsi heuristik. Fungsi heuristik adalah aturan-aturan yang digunakan untuk mendapatkan solusi yang diinginkan. Heuristik jarak ditambah biaya adalah penjumlahan dari dua fungsi, yaitu:

- fungsi jalur biaya, dimana biayanya dihitung dari node awal hingga node saat ini (biasanya dinotasikan $g(x)$).
- estimasi heuristik yang dapat diterima (admissible) dari jarak menuju goal (biasanya dinotasikan $h(x)$).

III. ANALISIS DAN PEMBAHASAN

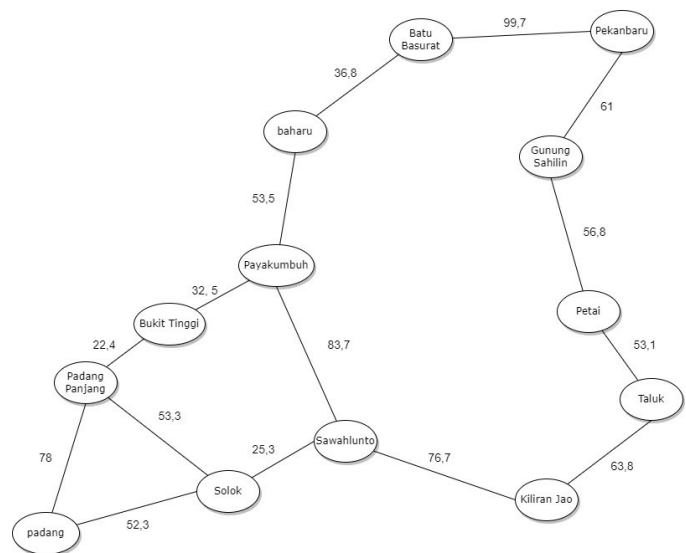
A. Visualisasi Graf pada Jalur Alternatif Padang-Pekanbaru

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



Gambar 3. Visualisasi Graf jalur Padang-Pekanbaru
Sumber : Google Maps

Pada gambar 3, terlihat titik-titik merah yang divisualisasikan sebagai simpul dari graf yang merupakan daerah jalur yang dapat dilalui dari Padang untuk menuju ke Pekanbaru. Selanjutnya penulis akan melakukan penyederhanaan terhadap gambar 3 menjadi sebagai berikut.



Gambar 4. Penyederhanaan Graf
Sumber : Dokumen Penulis

Visualisasi graf pada gambar 4 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 4 dapat dilihat berbagai jalur dan daerah yang dapat dilalui jika ingin pergi ke Pekanbaru dari Padang. Dengan demikian ini akan mempermudah penulis untuk menentukan rute terpendek dengan mengimplementasikan algoritma A*.

B. Penerapan Algoritma A* dalam menentukan Rute Terpendek Jalur Alternatif Padang-Pekanbaru

Algoritma A* digunakan untuk menentukan rute terpendek dari suatu tempat ke tempat lainnya. Algoritma A* akan menghasilkan solusi yang optimum. 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.

Dalam mengevaluasi beberapa jalur yang mungkin untuk menentukan jalur terpendek, algoritma A* menggunakan fungsi evaluasi. Rumus fungsi evaluasi yang digunakan adalah sebagai berikut.

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

$g(n)$ adalah jarak tempuh dari daerah awal ke suatu daerah tertentu di dalam graf. $g(n)$ 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 Pekanbaru yang telah penulis dapatkan dari google map menggunakan fasilitas (fitur) measure distance yang telah disediakan oleh google map. Nilai $h(n)$ didapatkan sebagai berikut.

Tabel 1. Nilai $h(n)$ setiap simpul

Simpul	$h(n)$ (dalam km)
Padang	198,1
Solok	166,5
Padang Panjang	157,1
Bukit Tinggi	147,5
Payakumbuh	119,7
Sawahlunto	147,87
Baharu	91,67
Kiliran Jao	154,4
Batu Basurat	82,3
Taluk	117,2
Petai	90,1
Gunung Sahilin	47,3

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 Kota Padang.
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 Padang ke Pekanbaru.

Tabel 2. Analisis Algoritma A*

Simpul Ekspan	Simpul Hidup	$f(n) = g(n) + h(n)$
Padang	Padang Panjang - Padang	$78 + 157,1 = 235,1$
	Solok - Padang	$52,3 + 166,5 = 218,8$
Solok - Padang	Sawahlunto – Solok - Padang	$77,6 + 147,87 = 225,47$
	Padang Panjang – Solok - Padang	$105,6 + 157,1 = 262,7$
	Padang Panjang – Padang	$78 + 157,1 = 235,1$
Sawahlunto –	Payakumbuh -	$161,3 + 119,7 =$

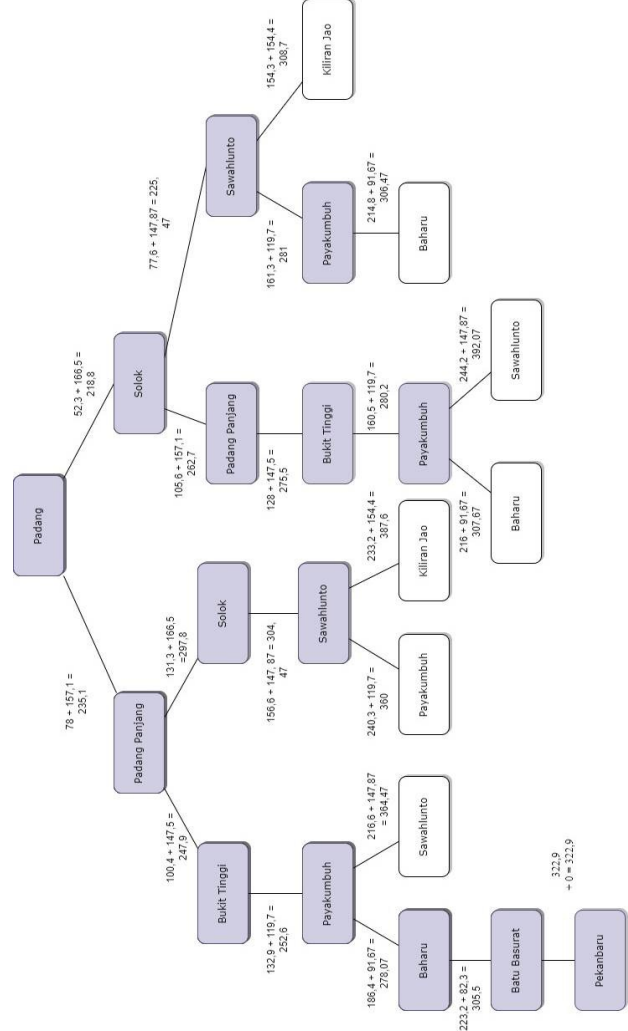
Solok- Padang	Sawahlunto – Solok- Padang	281
	Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$
	Padang Panjang – Solok - Padang	$105,6 + 157,1 = 262,7$
	Padang Panjang – Padang	$78 + 157,1 = 235,1$
Padang Panjang – Padang	Bukit Tinggi - Padang Panjang – Padang	$100,4 + 147,5 = 247,9$
	Solok - Padang Panjang – Padang	$131,3 + 166,5 = 297,8$
	Payakumbuh - Sawahlunto – Solok- Padang	$161,3 + 119,7 = 281$
	Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$
	Padang Panjang – Solok - Padang	$105,6 + 157,1 = 262,7$
Bukit Tinggi - Padang Panjang – Padang	Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$132,9 + 119,7 = 252,6$
	Solok - Padang Panjang – Padang	$131,3 + 166,5 = 297,8$
	Payakumbuh - Sawahlunto – Solok- Padang	$161,3 + 119,7 = 281$
	Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$
	Padang Panjang – Solok - Padang	$105,6 + 157,1 = 262,7$
Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$186,4 + 91,67 = 278,07$
	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$216,6 + 147,87 = 364,47$
	Solok - Padang Panjang – Padang	$131,3 + 166,5 = 297,8$
	Payakumbuh - Sawahlunto –	$161,3 + 119,7 = 281$

	Solok- Padang		Bukit Tinggi - Padang Panjang – Padang	Payakumbuh - Bukit Tinggi - Padang Panjang – Padang		
	Kiliran Jao - Sawahlunto – Solok- Padang	154,3 + 154,4 = 308,7		Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	160,5 + 119,7 = 280,2	
	Padang Panjang – Solok - Padang	105,6 + 157,1 = 262,7		Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	216,6 + 147,87 = 364,47	
Padang Panjang – Solok - Padang	Bukit Tinggi - Padang Panjang – Solok - Padang	128 + 147,5 = 275,5		Solok - Padang Padang Panjang – Padang	131,3 + 166,5 =297,8	
	Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	186,4 + 91,67 = 278,07		Payakumbuh - Sawahlunto – Solok- Padang	161,3 + 119,7 = 281	
	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	216,6 + 147,87 = 364,47		Kiliran Jao - Sawahlunto – Solok- Padang	154,3 + 154,4 = 308,7	
	Solok - Padang Padang Panjang – Padang	131,3 + 166,5 =297,8		Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	216 + 91,67 = 307,67
	Payakumbuh - Sawahlunto – Solok- Padang	161,3 + 119,7 = 281			Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	244,2 + 147,87 = 392,07
	Kiliran Jao - Sawahlunto – Solok- Padang	154,3 + 154,4 = 308,7			Batu Basurat - Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	223,2 + 82,3 = 305,5
	Bukit Tinggi - Padang Panjang – Solok - Padang	Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	160,5 + 119,7 = 280,2		Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	216,6 + 147,87 = 364,47
		Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	186,4 + 91,67 = 278,07		Solok - Padang Padang Panjang – Padang	131,3 + 166,5 =297,8
Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang		216,6 + 147,87 = 364,47	Payakumbuh - Sawahlunto – Solok- Padang	161,3 + 119,7 = 281		
Solok - Padang Padang Panjang – Padang		131,3 + 166,5 =297,8	Kiliran Jao - Sawahlunto – Solok- Padang	154,3 + 154,4 = 308,7		
Payakumbuh - Sawahlunto – Solok- Padang		161,3 + 119,7 = 281	Payakumbuh - Sawahlunto –	Baharu - Payakumbuh - Sawahlunto –	214,8 + 91,67 = 306,47	
Kiliran Jao - Sawahlunto – Solok- Padang		154,3 + 154,4 = 308,7				
Baharu - Payakumbuh -	Batu Basurat - Baharu -	223,2 + 82,3 = 305,5				

Solok- Padang	Solok- Padang		Padang Panjang – Padang			
	Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	$216 + 91,67 = 307,67$		Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$	
	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	$244,2 + 147,87 = 392,07$		Payakumbuh - Sawahlunto - Solok - Padang Panjang – Padang	$240,3 + 119,7 = 360$	
	Batu Basurat - Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$223,2 + 82,3 = 305,5$		Kiliran Jao - Sawahlunto - Solok - Padang Panjang – Padang	$233,2 + 154,4 = 387,6$	
	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$216,6 + 147,87 = 364,47$		Baharu - Payakumbuh - Sawahlunto – Solok- Padang	$214,8 + 91,67 = 306,47$	
	Solok - Padang Panjang – Padang	$131,3 + 166,5 = 297,8$		Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	$216 + 91,67 = 307,67$	
	Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$		Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	$244,2 + 147,87 = 392,07$	
	Solok - Padang Panjang – Padang	Sawahlunto - Solok - Padang Panjang – Padang		$156,6 + 147,87 = 304,47$	Sawahlunto - Solok - Padang Panjang – Padang	Batu Basurat - Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang
Baharu - Payakumbuh - Sawahlunto – Solok- Padang		$214,8 + 91,67 = 306,47$	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$216,6 + 147,87 = 364,47$		
Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang		$216 + 91,67 = 307,67$	Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$		
Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang		$244,2 + 147,87 = 392,07$	Batu Basurat - Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	Pekanbaru - Batu Basurat - Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang		$322,9 + 0 = 322,9$
Batu Basurat - Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang		$223,2 + 82,3 = 305,5$		Payakumbuh - Sawahlunto - Solok - Padang Panjang – Padang		$240,3 + 119,7 = 360$
Sawahlunto - Payakumbuh - Bukit Tinggi -		$216,6 + 147,87 = 364,47$		Kiliran Jao - Sawahlunto -		$233,2 + 154,4 = 387,6$

	Solok - Padang Panjang – Padang	
	Baharu - Payakumbuh - Sawahlunto – Solok- Padang	$214,8 + 91,67 = 306,47$
	Baharu - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	$216 + 91,67 = 307,67$
	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Solok - Padang	$244,2 + 147,87 = 392,07$
	Sawahlunto - Payakumbuh - Bukit Tinggi - Padang Panjang – Padang	$216,6 + 147,87 = 364,47$
	Kiliran Jao - Sawahlunto – Solok- Padang	$154,3 + 154,4 = 308,7$

Jika ditampilkan dalam graf akan menjadi



Gambar 5. Pohon Rute Terpendek menggunakan algoritma A*

Sumber : (dokumen penulis)

Dari gambar 5 disimpulkan bahwa jarak terpendek dari Kota Padang menuju Kota Pekanbaru adalah melalui rute Padang – Padang Panjang – Bukit Tinggi – Payakumbuh – Baharu – Batu Basurat – Pekanbaru dengan jarak 322,9 km

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. 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.

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 “Penentuan Rute terpendek dari Jalur Alternatif Padang - Pekanbaru dengan Menggunakan Graf dengan Algoritma A*”. 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 Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc., selaku pembimbing dan dosen K2. Karena tanpa beliau, penulis tidak akan mengerti tentang materi yang disampaikan pada makalah ini.

REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf> diakses pada 16 Mei 2022
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf> diakses pada 16 Mei 2022.
- [3] <https://sinta.unud.ac.id/uploads/wisuda/1108605031-3-bab2.pdf> diakses pada 16 Mei 2022
- [4] <https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf> diakses pada 17 Mei 2022

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, 19 Mei 2022



Ilham Pratama
13520041