# Penerapan Graf pada Google Maps untuk Pemilihan Rute Terdekat

Kelvin Rayhan Alkarim - 13521005[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
[1]*13521005@std.stei.itb.ac.id*

***ABSTRAKt*— Kemacetan lalu lintas sering kali menghambat pengemudi untuk mencapai tujuan tepat waktu. Untuk mengatasi masalah tersebut, diperlukan rute terpendek untuk meningkatkan efisiensi dan efektifitas waktu dalam melakukan mobilisasi dari satu tempat ke tempat lain. Untuk mendapatkan rute terpendek tersebut, diperlukan algoritma yang cocok agar rute yang ditemukan dapat mengatasi masalah kemacetan secara optimal. Dengan menggunakan teori graf, rute terpendek tersebut dapat ditemukan. Salah satu cara untuk menemukan rute terpendek (Shortest Path Graph) adalah menggunakan algoritma A star (A*).**

***Kata Kunci—graf, A* (Astar), google maps, rute terdekat, implementasi.***

## I. PENDAHULUAN

Pada zaman yang serba maju dan modern ini, teknologi komunikasi dan informasi adalah ilmu pengetahuan yang digunakan manusia untuk membantu menyelesaikan permasalahan demi mencapai tujuan. Telepon genggam adalah salah satu penerapan teknologi yang saat ini mengalami perkembangan yang cepat. Kini pengguna telepon genggam dapat membuka halaman web dimana pun dan kapan pun menggunakan jaringan internet. Selain telepon genggam, perkembangan teknologi yang pesat juga berdampak pada tingginya kecenderungan manusia untuk melakukan mobilisasi. Berkembangnya teknologi transportasi dan infrastruktur ini merupakan tuntutan dari manusia yang memiliki kebutuhan untuk berpindah tempat dengan waktu sesingkat mungkin Jumlah penduduk yang semakin bertambah menyebabkan masalah mobilisasi semakin kompleks, maka dibutuhkan suatu metode yang tepat untuk menghindari permasalahan yang dapat timbul seperti kemacetan dan kecelakaan..

Tingginya mobilisasi yang dilakukan manusia dapat memakan waktu yang banyak apabila memilih rute yang kurang efisien dan tidak memperhitungkan rute terpendek. Masalah utamanya adalah bagaimana cara untuk mencari rute terpendek dari suatu titik ke titik tujuan. Oleh karena itu, diperlukan cara untuk mendapatkan rute terpendek dari suatu titik ke titik tujuan agar tercipta efisiensi dan efektifitas waktu dalam melakukan mobilisasi.

Untuk mengatasi masalah tersebut, salah satu aplikasi yang ada pada telepon genggam yang saat ini banya digunakan adalah Google Maps. Aplikasi ini akan mendapatkan informasi loksi pengguna dari GPS dan selanjutnya pengguna akan memasukkan tujuan yang diinginkan. Google Maps akan mencari jalur terpendek yang efisien untuk pengguna. cara aplikasi ini mencari rute terpendek adlaah dengan menggunakan algoritma yang cocok agar rute yang ditemukan optimal dan akurat. Rute terpendek yang dinaksud dapat dicari menggunakan algoritma graf.

Algoritma graf adalah suatu metode atau langkah-langkah yang digunakan untuk menyelesaikan suatu permasalahan yang berkaitan dengan graf. Algoritma graf dapat digunakan untuk mencari solusi dari berbagai permasalahan yang terkait dengan graf, seperti mencari jalur terpendek antar dua simpul, mencari komponen terbesar dalam suatu graf, atau mencari jumlah siklus dalam suatu graf.

Graf dapat digunakan dalam berbagai situasi untuk menggambarkan hubungan antar elemen atau variable dalam data. Engineer dapat menggunakan algoritma graf dalam berbagai bidang, seperti dalam mendesain jaringan jalan atau jembatan, menentukan cara terbaik untuk mengirim barang dari satu lokasi ke lokasi lain, atau mencari rute terpendek dari satu tempat ke tempat lain. Algoritma graf juga dapat digunakan dalam perencanaan proyek, misalnya untuk memetakan dan mengelola tugas-tugas yang harus diselesaikan dalam suatu proyek serta menentukan urutan yang tepat untuk menyelesaikan tugas-tugas tersebut. Algoritma graf juga dapat digunakan dalam penjadwalan mesin atau peralatan untuk memastikan bahwa semua tugas dapat diselesaikan dalam waktu yang efisien

## II. LANDASAN TEORI

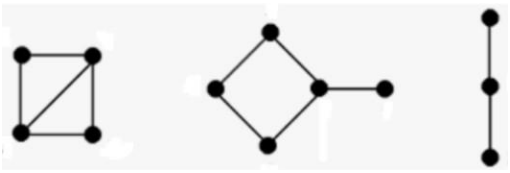
### 2.1 Graf

Graf adalah sebuah model matematis yang digunakan untuk merepresentasikan hubungan antara objek-objek atau node-node. Graf terdiri dari simpul-simpul atau node-node, yang mewakili objek-objek yang ingin ditunjukkan, dan sisi-sisi atau edges yang menunjukkan hubungan antara simpul-simpul tersebut. Jadi, simpul adalah titik-titik dalam graf, sedangkan sisi adalah garis-garis yang menghubungkan simpul-simpul tersebut. Jika graf di definisi kan sebagai G = (V, E) maka, V merupakan himpunan dari simpul - simpul ( { v1, v2, ... , vn} ) dan E merupakan himpunan dari sisi yang menghubungkan sepasang simpul tersebut ( {e1, e2, ... , en} ).

### 2.2 Jenis-jenis Graf

Graf dapat digolongkan menjadi beberapa jenis, yang pertama adalah graf sederhana (*simple graph*), yaitu graf yang tidak memiliki sisi yang menghubungkan simpul ke dirinya

sendiri (self-loop) dan tidak memiliki dua sisi yang menghubungkan dua simpul yang sama (multiple edges). Graf sederhana juga tidak memiliki simpul yang terisolasi atau tidak memiliki sisi yang menghubungkannya dengan simpul lainnya.

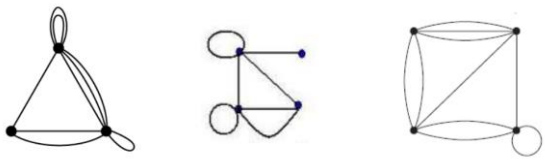


Graf Sederhana, sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

Yang kedua adalah graf tak-sederhana (*unsimple-graph*), yaitu graf yang memiliki sisi yang menghubungkan simpul ke dirinya sendiri (self-loop), memiliki dua sisi yang menghubungkan dua simpul yang sama (multiple edges), atau memiliki simpul yang terisolasi.

Ada beberapa jenis graf tak-sederhana, diantarnaya adalah:

a. Graf yang memiliki self-loop adalah graf yang memiliki sisi yang menghubungkan simpul ke dirinya sendiri. Misalnya, dalam graf berikut, simpul A memiliki self-loop karena terdapat sisi yang menghubungkan A ke A



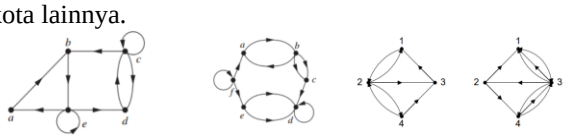
Graf Tak-Sederhana, sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

b. Graf yang memiliki multiple edges adalah graf yang memiliki dua sisi atau lebih yang menghubungkan dua simpul yang sama. Misalnya, dalam graf berikut, simpul A dan B memiliki multiple edges karena terdapat dua sisi yang menghubungkan A ke B

Graf Tak-Sederhana, sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

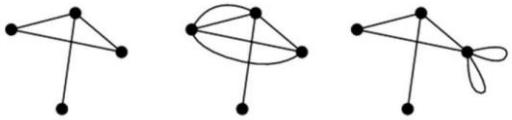
Graf juga dapat dibedakan berdasarkan orientasi arah pada sisi, yaitu: Graf tak-berarah (*undirected graph*) dimana sisi-sisi pada graf tidak mempunyai orientasi arah; Graf berarah (*directed graph* atau *digraph*) dimana setiap sisi-sisi pada graf diberikan orientasi arah.

a. Graf terarah adalah graf yang sisi-sisinya memiliki arah atau orientasi. Misalnya, graf terarah dapat digunakan untuk menggambarkan hubungan antara kota-kota dalam jaringan transportasi, dimana setiap sisi menunjukkan arah perjalanan dari satu kota ke kota lainnya.



Graf Berarah, sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

b. Graf tidak terarah adalah graf yang sisi-sisinya tidak memiliki arah atau orientasi. Misalnya, graf tidak terarah dapat digunakan untuk menggambarkan hubungan antar orang dalam sebuah jaringan sosial, dimana setiap sisi menunjukkan bahwa orang tersebut saling mengenal satu sama lain.



Graf Tak-Berarah, sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

2.3 Aplikasi Graf

Graf dapat digunakan dalam berbagai situasi untuk menggambarkan hubungan antar elemen atau variabel dalam data. Beberapa contoh penerapan graf adalah sebagai berikut:

1. Analisis jaringan sosial: Graf dapat digunakan untuk menggambarkan hubungan antar orang dalam sebuah jaringan sosial. Graf terarah dapat digunakan untuk menggambarkan hubungan pertemanan, dimana sisi menunjukkan bahwa orang tersebut adalah teman dari orang lainnya. Graf tidak terarah dapat digunakan untuk menggambarkan hubungan kekerabatan, dimana sisi menunjukkan bahwa orang tersebut saling berkaitan.

2. Analisis sistem transportasi: Graf dapat digunakan untuk menggambarkan hubungan antar kota dalam jaringan transportasi. Graf terarah dapat digunakan untuk menggambarkan arah perjalanan dari satu kota ke kota lainnya. Graf tidak terarah dapat digunakan untuk menggambarkan kota-kota yang dapat dicapai dengan berbagai cara transportasi.

3. Analisis aliran air: Graf dapat digunakan untuk menggambarkan hubungan antar daerah dalam suatu sistem irigasi. Graf terarah dapat digunakan untuk menggambarkan aliran air dari sumber air ke daerah pengguna air. Graf tidak terarah dapat digunakan untuk menggambarkan daerah-daerah yang saling terhubung dalam sistem irigasi tersebut.

4. Analisis jaringan elektronik: Graf dapat digunakan untuk menggambarkan hubungan antar perangkat dalam sebuah jaringan elektronik. Graf terarah dapat digunakan untuk menggambarkan aliran sinyal dari satu perangkat ke perangkat lainnya. Graf tidak terarah dapat digunakan untuk

menggambarkan perangkat-perangkat yang saling terhubung dalam jaringan elektronik tersebut.

Terdapat banyak lagi contoh penerapan graf dalam kehidupan sehari-hari, seperti dalam analisis jaringan telekomunikasi, jaringan pekerjaan, dan jaringan makanan. Graf dapat membantu kita memahami dan menganalisis data dengan lebih mudah dan efektif.

2.4 Algoritma Graf

Beberapa algortima graf yang umum digunakan adalah sebagai berikut:

a. Algoritma Dijkstra

Algoritma yang digunakan untuk mencari jalur terpendek antar dua simpul dalam suatu graf berbobot. Algoritma ini menggunakan teknik pemilihan simpul terbaik untuk mencari jalur terpendek yang memenuhi syarat.

b. Algoritma Prim

Algoritma yang digunakan untuk mencari komponen terbesar dalam suatu graf tidak terarah. Algoritma ini menggunakan teknik pemilihan sisi terbaik untuk mencari komponen terbesar yang memenuhi syarat.

c. Algoritma Kruskal

Algoritma yang digunakan untuk mencari komponen

## III. PERBANDINGAN METODOLOGI

Perbedaan dan perbandingan beberapa algoritma pencarian rute terpendek untuk menentukan algoritma mana yang cocok digunakan dalam pencarian rute terpendek Google Maps, dapat dilihat melalui parameter-parameter berikut:

### 3.1 Kompleksitas Waktu

Kompleksitas waktu merupakan aspek terkait waktu yang diperlukan untuk menemukan solusi dalam menyelesaikan suatu persoalan. Dalam menentukan kompleksitas ada beberapa variabel yang diperhitungkan, di antaranya adalah sebagai berikut,

O: merepresentasikan laju pertumbuhan suatu algoritma bergantung masukannya

|V|: node pada graph

|E|: edge pada graph

Pada beberapa algoritma yang telah dipilih, diperoleh informasi kompleksitas waktu dari algoritma tersebut adalah sebagai berikut,

A*: O(|E|)

BFS: O(|V|+|E|)

DFS: O(|V|+|E|)

Dijkstra: O(|V|+|V| log |V|) Floyd-Warshall: O(|V|3)

### 3.2 Kompleksitas Ruang

Kompleksitas ruang merupakan aspek terkait memori atau penyimpanan yang diperlukan dalam melakukan suatu pencarian solusi. Sama halnya seperti kompleksitas waktu dalam menentukan kompleksitas ada beberapa variabel yang diperhitungkan, yaitu sebagai berikut,

O: merepresentasikan laju pertumbuhan suatu algoritma bergantung masukannya

|V|: node pada graph

|E|: sisi pada graph

Pada beberapa algoritma yang telah dipilih, diperoleh informasi kompleksitas ruang dari algoritma tersebut adalah sebagai berikut,

A*: O(|V|)

BFS: O(|V|)

DFS: O(|V|)

Dijkstra: O(|V|)

Floyd-Warshall: $O(|V|^2)$

### 3.3 Keoptimalan

Keoptimalan merupakan aspek terkait dengan ditemukannya solusi yang optimal, yaitu pencarian berhasil menemukan solusi dengan tingkat efisiensi dan efektifitas yang terbaik. Pada beberapa algoritma yang

telah dipilih, diperoleh informasi keoptimalan dari algoritma tersebut adalah sebagai berikut,

A*: Iya, karena adanya fungsi heuristik yang meningkatkan efisiensi dan efektifitas.

BFS: Tidak, karena hasilnya sangat bergantung pada karakter masukan.

DFS: Kurang, sedikit lebih baik dari BFS tetapi masih kurang dibandingkan yang lain.

Dijkstra: Iya, karena adanya fungsi optimalisasi mirip pada algoritma A*.

Floyd-Warshall: Iya, karena algoritmanya yang dinamis jadi memungkinkan optimalisasi.

3.4 Ketuntasan

Ketuntasan merupakan aspek terkait dengan ditemukannya atau tidak suatu solusi yang diinginkan dalam penyelesaian persoalan. Pada beberapa algoritma yang telah dipilih, diperoleh informasi ketuntasan dari algoritma tersebut adalah sebagai berikut,

A*: Iya, karena optimalisasi yang sangat tinggi dengan adanya fungsi heuristik.

BFS: Cukup, memungkinkan solusi tetapi tidak selalu yang terbaik.

DFS: Cukup, memungkinkan solusi tetapi tidak selalu yang terbaik.

Dijkstra: Iya, adanya fungsi yang serupa pada A* menambah tingkat ketuntasan.

Floyd-Warshall: Iya, kedinamisan program jadinya memungkinkan pencarian lebih baik.

Sebagai kesimpulan, algoritma A* (A-Star) merupakan algoritma terbaik yang dapat digunakan untuk menyelesaikan masalah pencarian shortest path pada Google Maps.

## IV. METODOLOGI A-STAR

4.1 Perhitungan Algoritma A-Star

Algoritma A* (A-Star) menggunakan estimasi jarak terdekat (cost/ jarak sebenarnya) untuk mencapai tujuan (goal). Algoritma ini juga memiliki nilai heuristik yang digunakansebagai dasar pertimbangan pemilihan jalur. Algoritma A* memeriksa node dengan menggabungkan cost yang dibutuhkanuntuk mencapai sebuah node *g(n)* dan cost yang didapat dari node ke tujuan *h(n)*. Sehingga *f(n)* estimasi total biaya (cost) sebuah jalur (path) dari node awal ke node tujuan (goal) melaluinode n dapat dirumuskan sebagai berikut,

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

Untuk menentukan nilai h(n) ditunjukan oleh persamaan,

$$h(n) = (Xn-Xgoal)^2 + (Yn-Ygoal)^2$$

dimana,

*Xn* = nilai koordinat X dari node awal

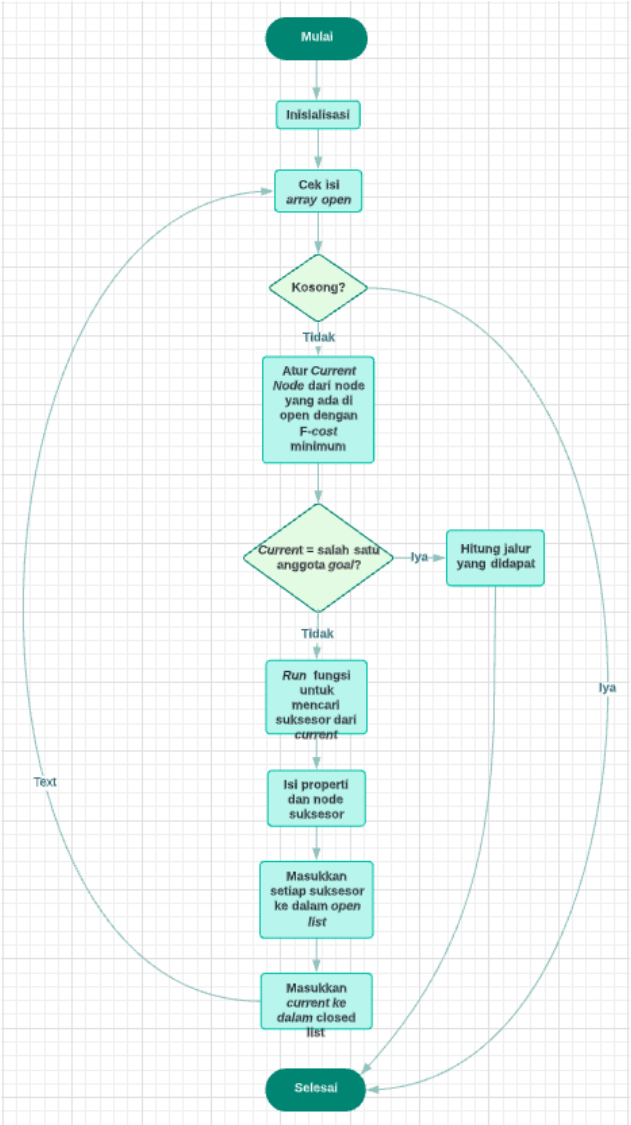
*Yn* = nilai koordinat Y dari node awal

*Xgoal* = nilai koordinat X dari node lokasi ke tujuan

*Ygoal* = nilai koordinat Y dari node lokasi ke tujuan

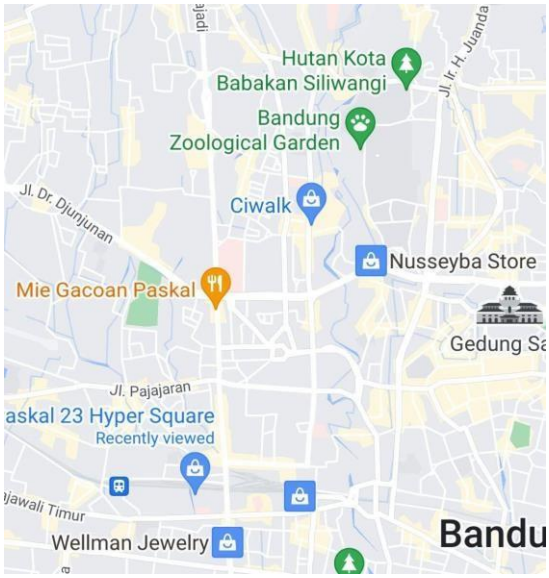
4.2 Flowchart Algoritma A-Star

Berikut merupakan diagram atau flowchart prinsip kerjaalgoritma A* dalam pencarian rute terpendek.

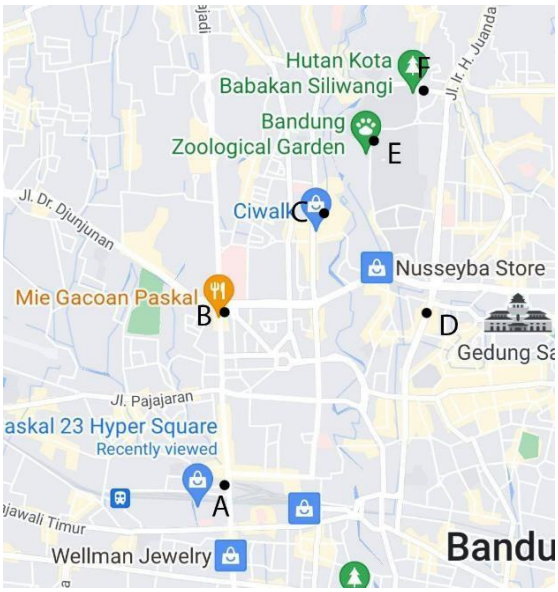


## V. IMPLEMENTASI

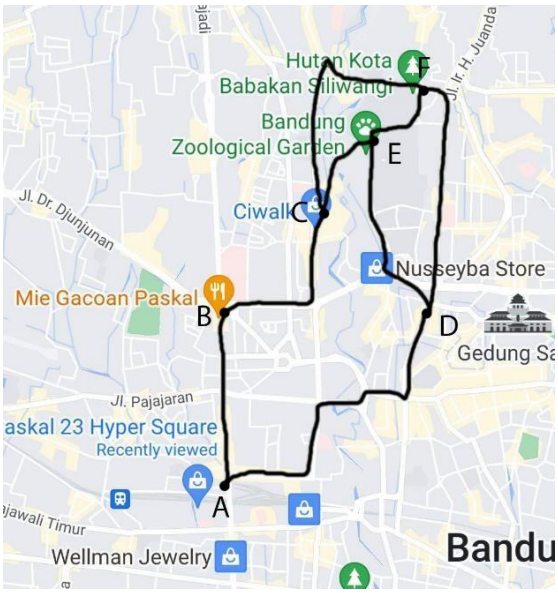
Berikut ini merupakan contoh simulasi penerapan algoritma A* untuk mencari jalan terdekat atau shortest path. Pada kasus kali ini, kami ingin mencari jalan terdekat dari Paskal 23 Hyper Square menuju ke Hutan Kota Babakan Siliwangi.



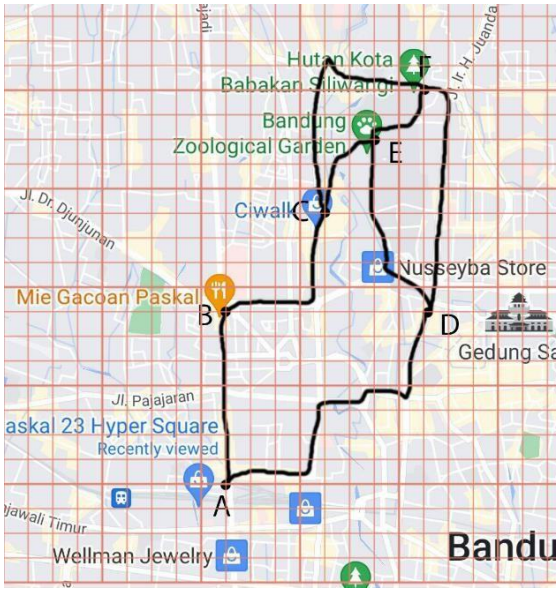

Asumsikan starting point (Paskal 23 Hyper Square) berada di titik A. Sedangkan end point (Hutan Kota Babakan Siliwangi) berada di titik F. Untuk titik-titik lainnya seperti B, C, D, dan E merupakan simpul-simpul yang menjadi opsi dalam pemilihan jalur.




Setiap simpul dihubungkan menggunakan garis hitam agar dapat terlihat rutenya dengan lebih jelas.




Maps pada Google Maps diberi garis koordinat seperti gambar berikut agar koordinat setiap simpul dapat dicari untuk



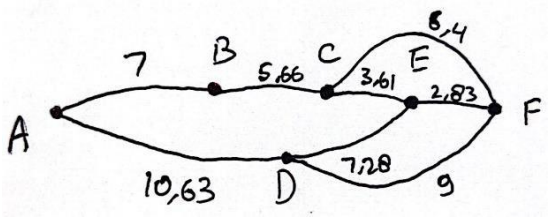

menghitung h(n). Asumsikan titik A berada di (0,0).

Titik koordinat:
A. (0,0)
B. (0,7)
C. (4,11)
D. (8,7)
E. (6,14)
F. (8,16)

Perhitungan bobot adalah sebagai berikut:
A ke B -> (0-0)2+(7-0)2 = 7
A ke D -> (8-0)2+(7-0)2 = 10.63
B ke C -> (4-0)2+(11-7)2 = 5.66
C ke F -> (8-4)2+(16-11)2 = 6.4
C ke E -> (6-4)2+(14-11)2 = 3.61
D ke E -> (6-8)2+(14-7)2 = 7.28
D ke F -> (8-8)2+(16-7)2 = 9

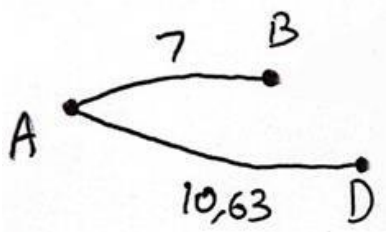
E ke F -> (8-6)2+(16-14)2 = 2.83

Ilustrasi graf:



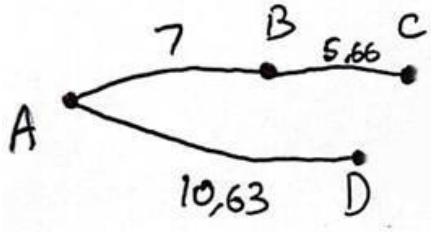

Langkah-langkah penyelesaian:

1. Karena di OPEN hanya terdapat 1 simpul (yaitu A), maka A Terpilih sebagai Best Node.



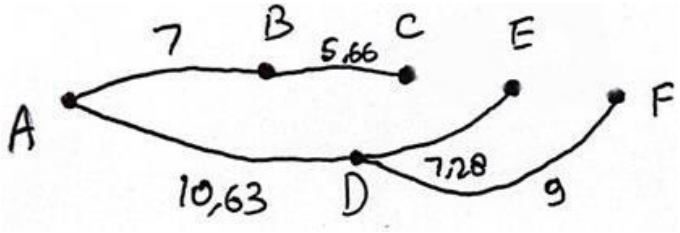

Closed: A Open: B, D f(B) = 0 + 7 = 7
f(D) = 0 + 10.63 = 10.63
Best Node selanjutnya adalah f(B) = 7

2. B dengan biaya terkecil (yaitu 7) terpilih sebagai Best Node dan dipindahkan ke CLOSED, semua suksesor B dibuka yaitu C, dan dimasukkan ke OPEN.



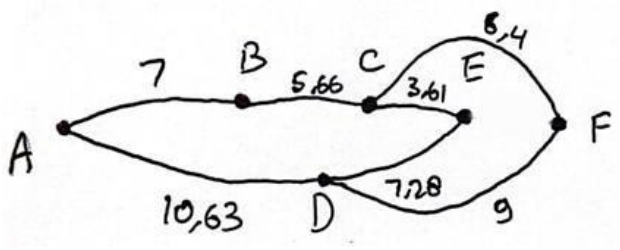

Closed: A, B Open: C, D
f(C) = 0 + 7 + 5.66 = 12.66
f(D) = 0 + 10.63 = 10.63
Best Node selanjutnya adalah f(D) = 10.63

3. D dengan biaya terkecil (yaitu 10.63) terpilih sebagai Best Node dan dipindahkan ke CLOSED, semua suksesor D dibuka yaitu E dan F, dan dimasukkan ke OPEN.



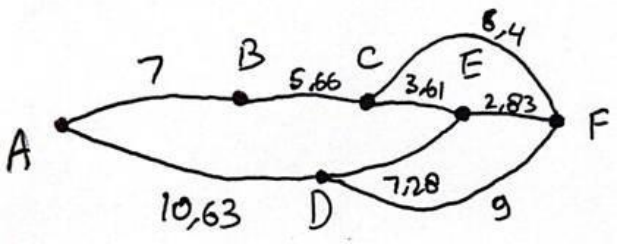

Closed: A, B, D Open: C, E, F
f(C) = 0 + 7 + 5.66 = 12.66
f(E) = 0 + 10.63 + 7.28 = 17.91
f(F) = 0 + 10.63 + 9 = 19.63
Best Node selanjutnya adalah f(C) = 12.66

4. C dengan biaya terkecil (yaitu 12.66) terpilih sebagai Best Node dan dipindahkan ke CLOSED, semua suksesor C dibuka yaitu E dan F, dan dimasukkan ke OPEN.




Closed: A, B, C, D Open: E, F
f(F) dari C = 0 + 7 + 5.66 + 6.4 = 19.06
f(E) dari C = 0 + 7 + 5.66 + 3.61 = 16.27
f(E) dari D = 0 + 10.63 + 7.28 = 17.91
f(F) dari D = 0 + 10.63 + 9 = 19.63
Best Node selanjutnya adalah f(E) dari C = 16.27

5. E dari C dengan biaya terkecil (yaitu 16.27) terpilih sebagai Best Node dan dipindahkan ke CLOSED, semua suksesor E dibuka yaitu F, dan dimasukkan ke OPEN.




Closed: A, B, C, D, E Open: F
f(F) dari C = 0 + 7 + 5.66 + 6.4 = 19.06
f(F) dari E, C = 0 + 7 + 5.66 + 3.61 + 2.83= 19.1
f(F) dari E, D = 0 + 10.63 + 7.28 + 2.83 = 20.74
f(F) dari D = 0 + 10.63 + 9 = 19.63

Dengan demikian, algoritma pencarian A-Star sudah selesai, dengan hasil rute terpendek A -> B -> C -> F dan jarak 19.06 satuan.

## VI. KESIMPULAN

Solusi yang dapat memudahkan pencarian rute terpendek (Shortest Path Graph) adalah dengan menggunakan berbagai macam algoritma graf seperti algoritma A star (A*), algoritma Djikstra, algoritma Floyd-Warshall dan lain-lainnya. Algoritma A* memiliki skor yang sangat baik sebagai solusi dari proses pencarian rute terpendek yang akan ditempuh suatu poin awal (starting point) sampai ke objek tujuan. Algoritma A* dapat diimplementasikan pada aplikasi Google Maps sebagai pencari rute terpendek dengan waktu yang relatif singkat dan biaya yang lebih hemat sehingga menjadi solusi pekerjaan dengan mobilitas tinggi.

## REFERENCES

[1] Needham dan Hodler, “Graph Algorithms”. 1st ed. United State of America: y O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2019.
[2] Munir, Rinaldi (2003). Graf (Bag.1) Bahan Kuliah IF2120 Matematika Diskrit, URL: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf- 2020-Bagian1.pdf
Diakses: 3 Desember 2022, pukul 1:51.
[3] Adiputra, Muhamad Arif.(2018, 14 Mei). Penerapan Algoritma BFS dan DFS untuk Penjadwalan Rencana Studi, URL: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Makalah/Makalah-IF2211-2018-040.pdf.
Diakses: 2 Desember 2022, pukul 23:13.
[4] Hartanto, Kevin (2020, 21 Juli). Analisis Perbandingan Algoritma Dijkstra dan Algoritma A* (A-Star) Dalam Pencarian Rute Terpendek (Studi Kasus: SPBU di Kota Medan). URL: https://repositori.usu.ac.id/bitstream/handle/123456789/27965/1614010 65.pdf?sequence=1&isAllowed=y.
Diakses: 3 Desember 2022, pukul 3:06.
[5] Prasetiyo, A. Fajar. (2015, 9 Desember). Penerapan Algoritma Shortest Path pada Google Maps. URL: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Makalah-Matdis-2015/Makalah-IF2120-2015-023.pdf.
Diakses: 3 Desember 2022, pukul 8:09.
[6] Roy, Baijayanta. (2019, 29 September). A-Star (A*) Search Algorithm. https://towardsdatascience.com/a-star-a-search-algorithm- eb495fb156bb.
Diakses: 3 Desember 2022, pukul 9:32.

## 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, 10 Desember 2022

Kelvin Rayhan Alkarim
13521005