

# Aplikasi Graf dalam Penentuan Rute Penerbangan Internasional

Daniel Mulia Putra Manurung - 13522043<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13522043@mahasiswa.itb.ac.id

**Abstrak**—Perkembangan zaman yang pesat, terutama dengan munculnya berbagai jenis pekerjaan dan kemajuan industri, telah meningkatkan kebutuhan mobilisasi manusia, terutama dalam mobilisasi antar negara. Dengan tingginya tingkat permintaan penerbangan internasional, dan dengan terbatasnya sumber daya yang dapat digunakan, tinggi biaya penerbangan internasional menjadi salah satu permasalahan yang muncul. Graf merupakan salah satu bentuk pendekatan matematis yang dapat digunakan untuk menentukan rute terbaik penerbangan internasional dari suatu negara ke negara lain, sesuai dengan biaya tiap penerbangan yang memungkinkan. Pada makalah ini, akan dibahas pemanfaatan graf, dan penerapan algoritma bidirectional search untuk menyelesaikan permasalahan dalam menentukan rute terbaik penerbangan internasional sesuai dengan biayanya.

**Kata Kunci**—Graf, Algoritma Bidirectional Search, Rute Penerbangan Internasional.

## I. PENDAHULUAN

Seiring dengan perkembangan zaman, dengan munculnya berbagai jenis pekerjaan dan bidang perkembangan industri, kebutuhan mobilisasi semakin meningkat. Dari kebutuhan mobilisasi antar bangunan, mobilisasi antar kota, hingga mobilisasi antar negara. Kebutuhan mobilisasi ini dapat dipenuhi dengan adanya transportasi antar negara. Transportasi yang paling sering digunakan adalah pesawat dengan mobilisasi penerbangan internasional.

Penerbangan internasional telah menjadi bagian integral dalam mobilisasi dan konektivitas global di era modern. Dengan pertumbuhan industri penerbangan yang sangat pesat, dan dengan banyaknya kebutuhan manusia untuk melakukan penerbangan internasional, kebutuhan untuk menentukan rute penerbangan yang efisien dan ekonomis menjadi sangat penting. Permasalahan ini akan diselesaikan dengan fokus kepada aplikasi teknik graf. Dengan memanfaatkan graf berbobot, kita dapat menemukan cara untuk menentukan rute optimal dengan mempertimbangkan factor biaya.

Dengan menggabungkan prinsip-prinsip teori graf dan dengan data analisis biaya perjalanan antar negara, akan dirancang sebuah model penerapan graf berbobot yang merepresentasikan konektivitas antar negara. Bobot-bobot pada grad akan merepresentasikan biaya penerbangan dari rute tersebut. Untuk menentukan rute paling optimal dari perjalanan dari satu negara ke negara lainnya akan digunakan algoritma Bidirectional Search.

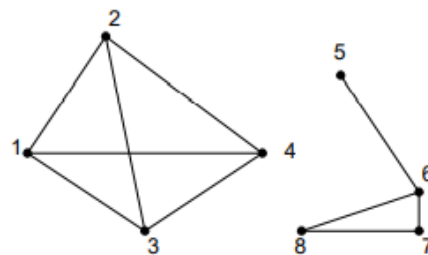
Tujuan dari penelitian ini adalah untuk menyajikan

pendekatan inovatif untuk dapat menentukan rute penerbangan internasional yang mempertimbangkan efisiensi biaya operasional, agar pelanggan dapat menentukan rencana mereka, sesuai dengan budget yang dimilikinya. Dengan menggunakan algoritma graf yang dioptimalkan, penelitian ini diharapkan untuk memberikan kontribusi pada pengembangan perencanaan rute penerbangan yang lebih ekonomis.

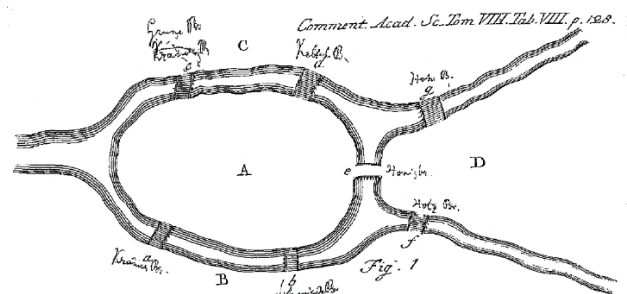
## II. LANDASAN TEORI

### 2.1 Graf

Graf atau juga dikenal sebagai “grafik” dalam konteks matematika dan ilmu computer merupakan struktur yang menghubungkan berbagai objek, di mana setiap objek memiliki keterkaitan tertentu dengan objek lainnya. Objek-objek pada graf direpresentasikan dengan node atau simpul, dan hubungan antar objek-objek direpresentasikan dengan edge atau sisi. [1]



Gambar 2.1 Contoh Graf



Gambar 2.2 Jembatan Königsberg

Graf dapat didefinisikan sebagai berikut,

$$G = (V, E)$$

V merepresentasikan himpunan simpul-simpul pada graf dan E merepresentasikan himpunan sisi-sisi yang menghubungkan simpul pada graf. Perlu diketahui bahwa V bukanlah himpunan kosong. Sehingga dengan demikian V dan E dapat dituliskan sebagai,

$$V = \{v_1, v_2, v_3, \dots, v_n\}$$

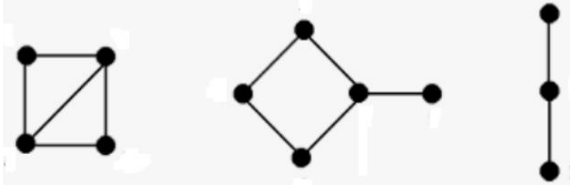
$$E = \{e_1, e_2, e_3, \dots, e_n\}$$

## 2.2 Jenis-Jenis Graf

Graf dapat digolongkan berdasarkan beberapa kriteria, berdasarkan ada tidaknya sisi ganda, atau gelang atau sisi yang menghubungkan sebuah simpul dengan dirinya sendiri. Graf ini terbagi menjadi graf sederhana dan graf tidak sederhana.

### 1. Graf Sederhana

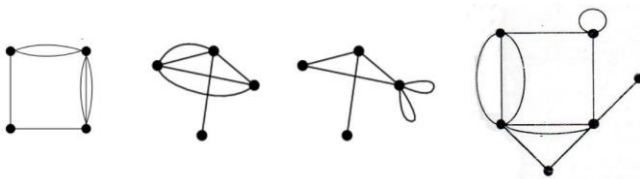
Graf sederhana adalah graf yang tidak mengandung gelang



Gambar 2.3 Graf Sederhana

### 2. Graf tidak sederhana

Graf tidak sederhana adalah graf yang memiliki sisi gelang atau sisi ganda.

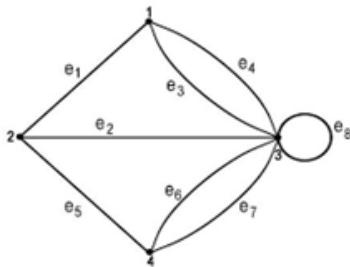


Gambar 2.4 Graf Tidak Sederhana

Graf juga dapat direpresentasikan dengan arah, yang disebut graf berarah. Sehingga berdasarkan orientasi arah pada graf, graf dapat dibedakan sebagai berikut, [2]

### 1. Graf tak berarah

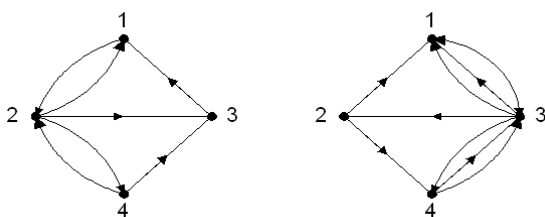
Graf tak berarah adalah graf yang tidak memiliki orientasi arah



Gambar 2.5 Graf Tak Berarah

### 2. Graf berarah

Graf berarah adalah graf yang memiliki orientasi arah



Gambar 2.6 Graf Berarah

## 2.3 Terminologi Graf

Graf memiliki beberapa terminologi penting, berikut adalah

beberapa diantaranya

### 1. *Adjacent* (Ketetanggaan)

Dua simpul pada graf  $G$  dapat dikatakan bertetangga bila kedua simpul tersebut terhubung langsung oleh sebuah sisi. Berdasarkan gambar 2.1, kita dapat melihat bahwa simpul 1 dan simpul 4 terhubung oleh sebuah sisi, maka simpul 1 dan simpul 4 dapat dikatakan bertetangga

### 2. *Incidency* (Bersisian)

Sebuah sisi dikatakan bersisian dengan simpul jika sisi tersebut menghubungkan simpul tersebut dengan simpul lain, dalam kata lain  $e = (v1, v2)$ , maka  $e$  bersisian dengan  $v1$  dan  $v2$ . Pada gambar 2.5 kita dapat melihat bahwa sisi  $e1$  bersisian dengan simpul 2 dan simpul 3.

### 3. *Null Graph* (Graf Kosong)

Graf kosong adalah graf yang himpunan sisinya ( $E$ ) adalah himpunan kosong, dengan kata lain tidak terdapat sisi yang menghubungkan simpul-simpul pada graf.



Gambar 2.7 Graf Kosong

### 4. *Degree* (Derajat)

Derajat sebuah simpul adalah jumlah sisi yang berurusan dengan simpul tersebut, atau dalam graf berarah, jumlah sisi keluar dari simpul tersebut. Pada gambar 2.1 kita dapat melihat bahwa terdapat tiga sisi yang bersisian dengan simpul 1, sehingga degree simpul 1 adalah 3.

### 5. *Path* (Lintasan)

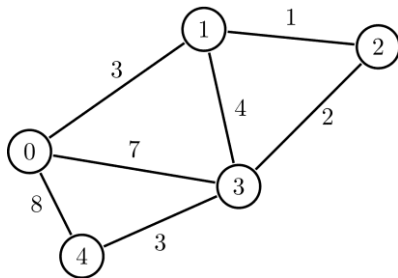
Lintasan adalah rangkaian sisi – sisi yang menghubungkan 2 simpul. Sebuah lintasan dikatakan memiliki Panjang  $n$  jika dari simpul yang pertama ke simpul akhir melewati  $n$  sisi. Pada gambar 2.1 kita dapat melihat bahwa lintasan  $1 - 4 - 2 - 3$  memiliki panjang  $n$ .

### 6. *Circuit* (Sirkuit)

Sirkuit adalah lintasan yang memiliki simpul awal dan simpul akhir yang sama, sehingga membentuk sebuah loop atau sirkuit. Kita dapat melihat pada gambar 2.1 bahwa lintasan  $1 - 4 - 3 - 2 - 1$  adalah sebuah sirkuit.

### 7. *Weighted Graph* (Graf Berbobot)

Graf berbobot adalah graf yang memiliki sisi yang telah diberi bobot. Bobot yang diberikan pada sisi graf dapat merepresentasikan banyak hal, tergantung keperluan implementasi graf.



Gambar 2.8 Graf Berbobot

### 2.3 Algoritma Bidirectional Search

Algoritma Bidirectional Search adalah pendekatan dalam algoritma pencarian yang melibatkan dua titik atau node dan melakukan pencarian dari arah yang berlawanan. Ide dasarnya adalah untuk menentukan titik tengah dan dengan kecepatan yang sebaik mungkin. Beberapa penerapan bidirectional search adalah menentukan jalur terpendek dalam sebuah graf, dengan prasyarat dua titik graf yang diinginkan telah diketahui. [3]

[4] Langkah – langkah melakukan bidirectional search pada graf berbobot adalah sebagai berikut:

1. Menentukan titik awal dan titik akhir yang ingin ditemukan rute terfisionnya.
2. Asumsikan jarak antar kedua titik sebagai tak terhingga.
3. Masukkan set simpul yang akan menjadi awal dari pencarian.
4. Lakukan pencarian simpul secara bersamaan, dengan melakukan secara terpisah dan memilih simpul – simpul dengan total biaya terkecil. Lakukan beberapa kali iterasi.
5. Untuk setiap iterasi, periksa apakah simpul yang dieksplorasi dari titik awal dan titik akhir bertemu, dan jika benar bertemu maka jalur terfision telah ditemukan.
6. Setelah pertemuan terjadi, lakukan konstruksi ulang dari titik awal hingga titik akhir berdasarkan informasi yang telah ditemukan.
7. Jalur dari titik awal ke titik akhir yang anda inginkan telah ditemukan.

## III. PEMBAHASAN

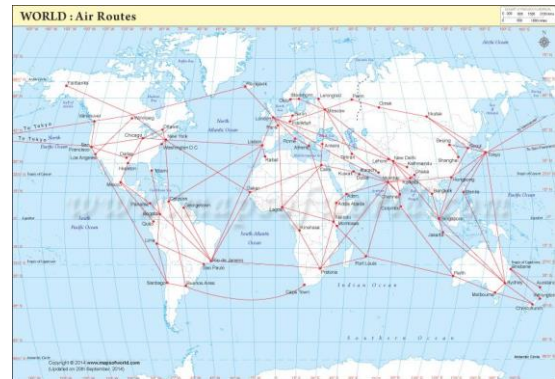
Rute paling optimal dalam penerbangan internasional dapat ditemukan menggunakan teori graf. Dengan menggunakan graf berbobot, rute paling efisien dapat didapatkan dengan berbagai algoritma yang dapat diterapkan dalam graf, dalam kasus ini algoritma yang digunakan adalah algoritma bidirectional search. Tujuan digunakannya algoritma bidirectional search adalah untuk mempercepat pencarian yang bersifat brute force. Objektif yang akan dicapai adalah untuk menentukan rute paling optimal dan efisien dari sebuah negara ke negara yang lain berdasarkan biaya yang diperlukan dalam perjalanannya.

Untuk menentukan graf yang akan digunakan, akan digunakan peta rute penerbangan internasional yang akan dimodifikasi menjadi graf yang lebih sederhana untuk mempermudah visualisasi. Modifikasi yang akan dilakukan adalah memperkecil range rute menjadi hanya melingkupi penerbangan internasional antar ibu kota utama di benua Asia Tenggara, Asia Timur, dan Australia. Kota – kota yang terlibat adalah Jakarta, Singapore, Sydney, Melbourne, Manila, Bangkok, Hongkong,

Tokyo, Beijing, dan Shanghai.

Untuk menentukan bobot dari graf, akan digunakan biaya rata-rata biaya dari tiap penerbangan internasional, dan untuk meningkatkan akurasi dari penggunaan algoritma, biaya penerbangan yang dipertimbangkan hanyalah biaya penerbangan langsung atau *direct*, dan bukan penerbangan yang melibatkan transit.

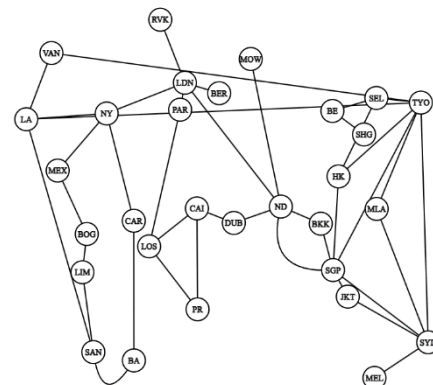
Berikut adalah rute penerbangan internasional yang belum dimodifikasi.



Gambar 3.1 Rute Penerbangan Internasional

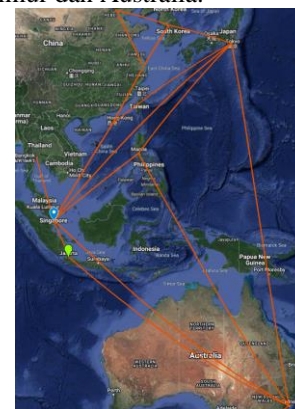
(sumber <https://www.mapsofworld.com/world-airroutes-map.htm> diakses pada 9 Desember 2023 pukul 22.13)

Dengan representasi graf sebagai berikut



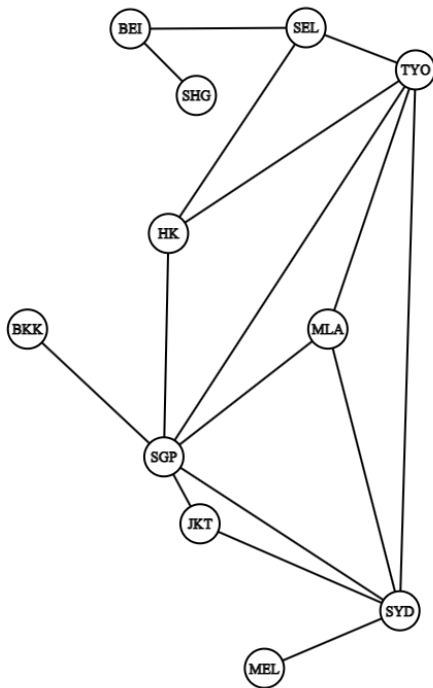
Gambar 3.2 Representasi Graf Rute Penerbangan Internasional

Untuk mempermudah visualisasi, rute penerbangan internasional akan dimodifikasi menjadi melingkupi Asia Tenggara, Asia Timur dan Australia.



Gambar 3.3 Rute Penerbangan Internasional Utama Asia Tenggara, Asia Timur, dan Australia

Dengan representasi graf sebagai berikut



Gambar 3.4 Representasi Graf Rute Penerbangan Internasional Utama Asia Tenggara, Asia Timur, dan Australia

Bobot graf akan didasarkan pada rata-rata biaya penerbangan internasional antar negara. Data biaya penerbangan internasional adalah sebagai berikut, [4]

JKT – SYD	Rp 7.620.000
JKT – SGP	Rp 1.600.000
SGP – SYD	Rp 6.408.000
SYD – MEL	Rp 2.932.000
SGP – BKK	Rp 1.979.000
SGP – MLA	Rp 2.465.000
SGP – TYO	Rp 5.486.000
SGP – HK	Rp 3.011.000
TYO – HK	Rp 4.500.000
TYO – SYD	Rp 16.104.000
SYD – MLA	Rp 7.969.000
MLA – TYO	Rp 3.841.000
HK – SEL	Rp 3.100.000
TYO – SEL	Rp 3.376.000
SEL – BEI	Rp 3.400.000
BEI – SHG	Rp 4.033.000

Untuk mempercepat proses iterasi dari bidirectional yang diimplementasikan, akan digunakan bahasa pemrograman python. Dengan bantuan program python, ditemukan jalur paling efisien dengan data yang telah diberikan, algoritma bidirectional search program python yang digunakan adalah sebagai berikut,

a. Import dan inialisasi Graf.

```

from collections import defaultdict
import heapq

class Graph:
    def __init__(self):
        self.graph = defaultdict(list)

    def add_edge(self, node1, node2, weight):
        self.graph[node1].append((node2, weight))
        self.graph[node2].append((node1, weight))
    
```

b. Algoritma pencarian bidirectional search

```

def bidirectional_search(graph, start, goal):
    forward = [(0, start, [])]
    backward = [(0, goal, [])]

    forward_discovered = {start: (0, [])}
    backward_discovered = {goal: (0, [])}

    common_node = None

    while forward and backward:
        forward_cost, forward_node, forward_path = heapq.heappop(forward)
        backward_cost, backward_node, backward_path = heapq.heappop(backward)

        if forward_node in backward_discovered:
            common_node = forward_node
            break

        for neighbor, cost in graph.graph[forward_node]:
            total_cost = forward_cost + cost
            if neighbor not in forward_discovered or total_cost < forward_discovered[neighbor][0]:
                forward_discovered[neighbor] = (total_cost, forward_path + [forward_node])
                heapq.heappush(forward, (total_cost, neighbor, forward_path + [forward_node]))

        for neighbor, cost in graph.graph[backward_node]:
            total_cost = backward_cost + cost
            if neighbor not in backward_discovered or total_cost < backward_discovered[neighbor][0]:
                backward_discovered[neighbor] = (total_cost, backward_path + [backward_node])
                heapq.heappush(backward, (total_cost, neighbor, backward_path + [backward_node]))

    if common_node is not None:
        forward_path = forward_discovered[common_node][1] + [common_node]
        backward_path = backward_discovered[common_node][1][::-1]
        return forward_path + backward_path
    else:
        return None
    
```

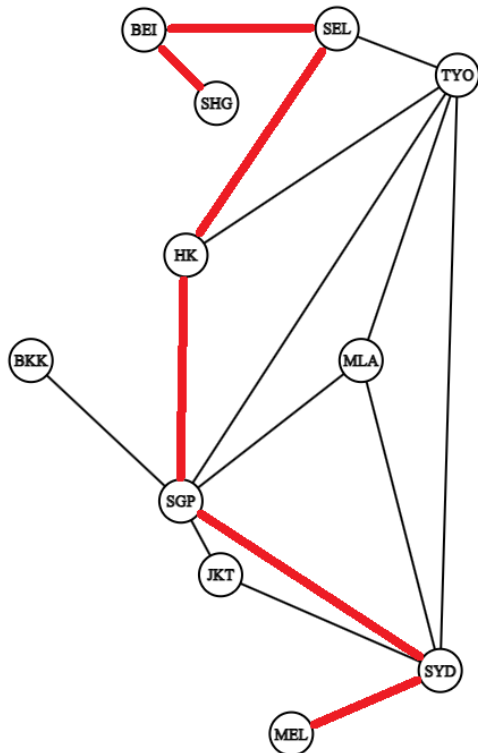
Dengan memasukkan data graf sesuai dengan biaya rata-rata penerbangan, Akan dicoba untuk mencari beberapa sample lokasi.

Pencarian pertama adalah pencarian rute penerbangan dari Melbourne ke Shanghai. Setelah program dijalankan, didapatkan hasil sebagai berikut:

Path from MEL to SHG: ['MEL', 'SYD', 'SGP', 'HK', 'SEL', 'BEI', 'SHG']

Gambar 3.5 Hasil Pencarian Rute Melbourne ke Shanghai

Kita mendapatkan hasil pencarian rute Melbourne ke Shanghai dengan melalui rute Melbourne, Sydney, Singapore, Hongkong, Seoul, Beijing, dan Shanghai. Representasi graf dapat dilihat sebagai berikut.



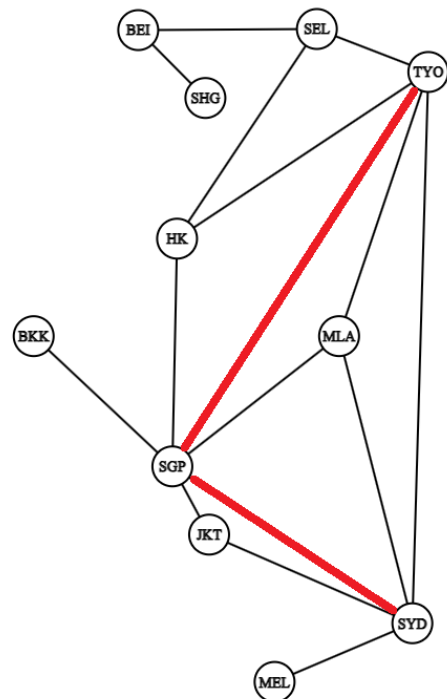
Gambar 3.6 Graf Hasil Pencarian Rute Melbourne ke Shanghai

Contoh lain dengan perbandingan lain yang dapat lebih mudah untuk dibandingkan adalah penerbangan Sydney ke Tokyo, setelah program dijalankan, hasilnya adalah sebagai berikut.

Path from SYD to TYO: ['SYD', 'SGP', 'TYO']

Gambar 3.7 Hasil Pencarian Rute Sydney ke Tokyo

Kita mendapatkan hasil pencarian rute Sydney ke Tokyo dengan melalui rute Sydney, Singapore, dan Tokyo. Representasi graf dapat dilihat sebagai berikut.



Gambar 3.8 Graf Hasil Pencarian Rute Sydney ke Tokyo.

Disini terlihat jelas bahwa rute tersebut tidak mengambil rute direct Sydney ke Tokyo. Hal ini dikarenakan rute Sydney direct ke Tokyo berbiaya Rp16.104.000,00 sedangkan rute Sydney ke Tokyo melalui Singapore secara akumulatif berbiaya Rp11.894.000,00.

Dengan demikian, algoritma bidirectional search dapat digunakan untuk mencari rute terbaik penerbangan internasional.

## V. KESIMPULAN

Solusi yang dapat digunakan untuk menentukan rute penerbangan internasional untuk memperkecil biaya, salah satunya adalah mengimplementasikan algoritma searching bidirectional pada teori graf. Dikarenakan rute penerbangan internasional sangatlah banyak dan beragam, untuk direpresentasikan dalam bentuk graf akan menghasilkan graf yang sangat kompleks.

## REFERENCES

- [1] Rinaldi Munir. 2023. Graf (Bag.1). <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf> [Diakses pada 10 Desember 2023 pukul 19.10]
- [2] *Graph theory in discrete mathematics - javatpoint*. www.javatpoint.com. (n.d.). <https://www.javatpoint.com/graph-theory-in-discrete-mathematics>
- [3] Point-to-point shortest path problem (P2P): Given: Efficient Point-to ... (n.d.). <https://www.cs.princeton.edu/courses/archive/spr06/cos423/Handouts/EP%20shortest%20path%20algorithms.pdf> [Diakses pada 10 Desember 2023 pukul 20.33]
- [4] GeeksforGeeks. (2023, October 8). *Bidirectional search*. GeeksforGeeks. <https://www.geeksforgeeks.org/bidirectional-search/> [Diakses pada 10 Desember 2023 pukul 22.43]

## 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 2023

A handwritten signature in black ink, consisting of several overlapping, fluid strokes that form a unique, stylized representation of the author's name.

Daniel Mulia Putra Manurung

13522043