

**MENGANALISIS *SHORTEST PATH* MENGGUNAKAN ALGORITMA  
FLOYD WARSHALL DAN ALGORITMA DIJKSTRA PADA  
DISTRIBUSI BARANG DARI KOTA-KOTA  
DI PULAU JAWA MENUJU KOTA  
BANDAR LAMPUNG**

**Skripsi**

**Oleh**

**AMRI SATRIAWAN ASH SHODIQ  
NPM. 2117031015**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2025**

## ABSTRACT

### ANALYZING THE SHORTEST PATH USING THE FLOYD WARSHALL ALGORITHM AND THE DIJKSTRA ALGORITHM ON THE DISTRIBUTION OF GOODS FROM CITIES ON THE ISLAND OF JAVA TO THE CITY OF BANDAR LAMPUNG

By

**Amri Satriawan Ash Shodiq**

The rapid growth of e-commerce has increased the demand for optimal goods distribution systems. One critical problem in logistics distribution is determining the shortest path and minimum distance from multiple source locations to a single destination. This study aims to analyze the performance of the Floyd–Warshall algorithm and the Dijkstra algorithm in solving the shortest path problem for goods distribution from sorting centers located in major and medium-sized cities on Java Island to a sorting center in Bandar Lampung. The distribution network is modeled as a weighted graph, where vertices represent cities and edge weights represent distances measured in kilometers. Data processing and computation are carried out using the Python programming language. The results show that the Floyd–Warshall algorithm is capable of determining the shortest distances between all pairs of vertices in a single computation and provides higher accuracy in distance calculation, but it does not directly present the routes taken. In contrast, the Dijkstra algorithm produces complete outputs, including the shortest distance, selected route, and visual representation for each source–destination pair, although it requires repeated computations and, in certain cases, yields longer distances compared to Floyd–Warshall. These findings indicate that Floyd–Warshall is more suitable for global distance analysis, while Dijkstra is more effective for route visualization and individual path determination.

**Keywords:** shortest route, Floyd Warshall algorithm, Dijkstra algorithm, distribution of goods.

## ABSTRAK

### MENGANALISIS *SHORTEST PATH* MENGGUNAKAN ALGORITMA FLOYD WARSHALL DAN ALGORITMA DIJKSTRA PADA DISTRIBUSI BARANG DARI KOTA-KOTA DI PULAU JAWA MENUJU KOTA BANDAR LAMPUNG

Oleh

**Amri Satriawan Ash Shodiq**

Pertumbuhan yang cepat pasar *online* meningkatkan kebutuhan untuk distribusi barang yang optimal. Salah satu masalah utama distribusi barang adalah menentukan *shortest path* dan jarak minimum dari banyak titik sumber menuju suatu titik tujuan. Skripsi ini bertujuan untuk menganalisis performa dari algoritma Floyd-Warshall dan Dijkstra dalam menyelesaikan masalah *shortest path* untuk distribusi barang dari pusat sortir yang ada di kota besar dan sedang yang ada di Pulau Jawa menuju pusat sortir di Bandar Lampung. Jaringan distribusi dimodelkan sebagai graf berbobot, dimana titik merepresentasikan kota dan bobot sisi merepresentasikan jarak dalam kilometer. Perhitungan data dan prosesnya menggunakan bahasa pemrograman Python. Hasilnya menunjukkan bahwa algoritma Floyd-Warshall mampu menentukan jarak terpendek dari setiap dua pasang titik dalam satu perhitungan dan memiliki akurasi yang lebih tinggi dalam perhitungan jarak, tetapi tidak secara langsung menunjukkan rute yang diambil. Sebaliknya, algoritma Dijkstra hasilnya lebih lengkap, yaitu jarak terpendek, rute yang diambil, dan peta visual dari setiap pasang dari titik sumber ke titik tujuan, meskipun diperlukan perhitungan berulang dan dalam kasus tertentu menghasilkan jarak yang lebih jauh dibanding Floyd-Warshall. Hal tersebut menunjukkan bahwa Floyd-Warshall lebih efektif untuk analisis jarak secara keseluruhan graf, sedangkan Dijkstra lebih efektif untuk visualisasi rute dan penentuan rute individu dalam graf.

**Kata-kata kunci:** rute terpendek, algoritma Floyd Warshall, algoritma Dijkstra, distribusi barang.

**MENGANALISIS *SHORTEST PATH* MENGGUNAKAN ALGORITMA  
FLOYD WARSHALL DAN ALGORITMA DIJKSTRA PADA  
DISTRIBUSI BARANG DARI KOTA-KOTA  
DI PULAU JAWA MENUJU KOTA  
BANDAR LAMPUNG**

**AMRI SATRIAWAN ASH SHODIQ**

**Skripsi**

Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2025**

Judul Skripsi : **MENGANALISIS *SHORTEST PATH*  
MENGUNAKAN ALGORITMA FLOYD  
WARSHALL DAN ALGORITMA  
DIJKSTRA PADA DISTRIBUSI BARANG  
DARI KOTA-KOTA DI PULAU JAWA ME-  
NUJU KOTA BANDAR LAMPUNG**

Nama Mahasiswa : **Amri Satriawan Ash Shodiq**

Nomor Pokok Mahasiswa : **2117031015**

Program Studi : **Matematika**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**



*Dorrah*  
**Dra. Dorrah Aziz, M.Si.**  
NIP 198406272006042001

*Agus Sutrisno*  
**Dr. Agus Sutrisno S.Si. M.Si.**  
NIP 198002062003121003

2. Ketua Jurusan Matematika

*Dr. Aang Nuryaman*  
**Dr. Aang Nuryaman, S.Si., M.Si.**  
NIP. 197403162005011001

**MENGESAHKAN**

1. tim penguji

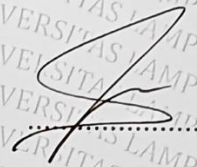
Ketua

: **Dra. Dorrah Aziz, M.Si.**



Sekretaris

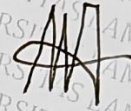
: **Dr. Agus Sutrisno S.Si. M.Si.**



Penguji

Bukan Pembimbing

: **Dr. Notiragayu S.Si. M.Si.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng. Heri Satria, S.Si., M.Si.**

NIP. 197110012005011002



**Tanggal Lulus Ujian Skripsi: 11 Desember 2025**

## PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : **Amri Satriawan Ash Shodiq**  
Nomor Pokok Mahasiswa : **2117031015**  
Jurusan : **Matematika**  
Judul Skripsi : **Menganalisis *Shortest Path* Menggunakan Algoritma Floyd Warshall dan Algoritma Dijkstra Pada Distribusi Barang Dari Kota-Kota di Pulau Jawa Menuju Kota Bandar Lampung**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung,

Penulis,



Amri Satriawan Ash Shodiq

## **RIWAYAT HIDUP**

Penulis memiliki nama lengkap Amri Satriawan Ash Shodiq yang lahir di Terbanggi Besar, Lampung Tengah pada tanggal 21 September 2003. Anak ketiga dari Bapak Sulani dan Ibu Husnul.

Penulis menempuh pendidikan di TK Miftahul Huda, dilanjutkan ke MI Miftahul Huda Nambah Dadi kemudian pindah ke SD Insan Kamil Bandar Jaya dan lulus tahun 2015. Sekolah Menengah Pertama di MTs Negeri 2 Bandar Lampung lulus pada tahun 2018. Sekolah Menengah Atas di SMA Negeri 1 Terbanggi Besar lulus pada tahun 2021. Penulis terdaftar sebagai mahasiswa S1 Jurusan Matematika Universitas Lampung pada tahun 2021 melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjadi mahasiswa penulis aktif di organisasi Natural pada tahun 2022-2023.

Penulis melaksanakan Kerja Praktek di Badan Pusat Statistik (BPS) Kabupaten Lampung Tengah pada Desember 2023-Februari 2024. Kemudian melaksanakan Kuliah Kerja Nyata (KKN) pada bulan Juni-Agustus 2024 di desa Kedaton Induk, Batanghari Nuban, Lampung Timur.

## **KATA INSPIRASI**

*"wa al laisa lil-insāni illā mā sa'ā, wa anna sa'yah saufa yurā"*  
(QS. An-Najm: 39-40)

*"Seseorang adalah protagonis dalam kehidupannya sendiri, fokus pada diri sendiri dan jangan bandingkan hidupmu dengan orang lain, teruslah berjalan kedepan dan lihatlah kebelakang untuk pengalaman"*

*"Alon-alon waton kelakon"*

## **PERSEMBAHAN**

Dengan mengucapkan Alhamdulillah dan syukur kepada Allah SWT atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan rasa syukur dan Bahagia, saya persembahkan rasa terimakasih saya kepada:

### **Ayah dan Ibuku Tercinta**

Terimakasih kepada orang tuaku atas segala pengorbanan, motivasi, doa dan ridho serta dukungannya selama ini. Terimakasih telah memberikan pelajaran berharga kepada anakmu ini tentang makna perjalanan hidup yang sebenarnya sehingga kelak bisa menjadi orang yang bermanfaat bagi banyak orang.

### **Dosen Pembimbing dan Pembahas**

Terimakasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga.

### **Sahabat-sahabatku**

Terimakasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasi, serta doa-doanya dan senantiasa memberikan dukungan dalam hal apapun.

### **Almamater Tercinta**

Universitas Lampung

## SANWACANA

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "Menganalisis *Shortest Path* Menggunakan Algoritma Floyd Warshall dan Algoritma Dijkstra Pada Distribusi Barang Dari Kota-Kota di Pulau Jawa Menuju Kota Bandar Lampung" dengan baik dan lancar serta tepat pada waktu yang telah ditentukan. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW.

Dalam proses penyusunan skripsi ini, banyak pihak yang telah membantu memberikan bimbingan, dukungan, arahan, motivasi serta saran sehingga skripsi ini dapat terselesaikan. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Ibu Dra. Dorrah Aziz, M.Si. selaku Pembimbing 1 yang telah banyak meluangkan waktunya untuk memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Dr. Agus Sutrisno S.Si. M.Si. selaku Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Ibu Dr. Notiragayu S.Si. M.Si. selaku Penguji yang telah bersedia memberikan kritik dan saran serta evaluasi kepada penulis sehingga dapat menjadi lebih baik lagi.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Bapak Prof. Ir. Netti Herawati, M.Sc., Ph.D. selaku dosen pembimbing akademik.

6. Seluruh dosen, staff dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Semoga skripsi ini dapat bermanfaat bagi kita semua. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menjadikan skripsi ini lebih baik lagi.

Bandar Lampung,

Amri Satriawan Ash Shodiq

## DAFTAR ISI

|  |             |
|--|-------------|
| <b>DAFTAR ISI</b> . . . . .                              | <b>xiii</b> |
| <b>DAFTAR TABEL</b> . . . . .                            | <b>xiv</b>  |
| <b>DAFTAR GAMBAR</b> . . . . .                           | <b>xv</b>   |
| <b>I PENDAHULUAN</b> . . . . .                           | <b>1</b>    |
| 1.1 Latar Belakang Masalah . . . . .                     | 1           |
| 1.2 Rumusan Masalah . . . . .                            | 2           |
| 1.3 Batasan Masalah . . . . .                            | 2           |
| 1.4 Tujuan Penelitian . . . . .                          | 2           |
| 1.5 Manfaat Penelitian . . . . .                         | 2           |
| <b>II TINJAUAN PUSTAKA</b> . . . . .                     | <b>3</b>    |
| 2.1 Transportasi . . . . .                               | 3           |
| 2.2 Teori Graf . . . . .                                 | 3           |
| 2.3 <i>Shortest Path Problem</i> . . . . .               | 6           |
| 2.3.1 Algoritma Floyd-Warshall . . . . .                 | 6           |
| 2.3.2 Algoritma Dijkstra . . . . .                       | 13          |
| <b>III METODE PENELITIAN</b> . . . . .                   | <b>15</b>   |
| 3.1 Waktu dan Tempat Penelitian . . . . .                | 15          |
| 3.2 Langkah Penelitian . . . . .                         | 15          |
| 3.2.1 Pengumpulan dan Pemahaman Materi Terkait . . . . . | 15          |
| 3.2.2 Pengumpulan Data . . . . .                         | 15          |
| 3.2.3 Membuat Graf . . . . .                             | 15          |
| 3.2.4 Perhitungan Menggunakan Program . . . . .          | 16          |
| 3.2.5 Analisa . . . . .                                  | 16          |
| 3.2.6 Menarik Kesimpulan . . . . .                       | 16          |
| <b>IV HASIL DAN PEMBAHASAN</b> . . . . .                 | <b>17</b>   |
| 4.1 Data . . . . .                                       | 17          |
| 4.2 Hasil dan Pembahasan . . . . .                       | 18          |
| 4.2.1 Algoritma Floyd Warshall . . . . .                 | 18          |
| 4.2.2 Algoritma Dijkstra . . . . .                       | 23          |

|   |           |
|---|-----------|
| <b>V KESIMPULAN DAN SARAN . . . . .</b> | <b>29</b> |
| 5.1 Kesimpulan . . . . .                | 29        |
| 5.2 Saran . . . . .                     | 30        |

## DAFTAR GAMBAR

|      |   |    |
|------|---|----|
| 2.1  | Contoh Graf (G) . . . . .   | 4  |
| 2.2  | Contoh Graf Berarah . . . . .   | 5  |
| 2.3  | Contoh Graf berbobot . . . . .  | 5  |
| 2.4  | Contoh Floyd-Warshall Graf (G) . . . . .  | 7  |
| 2.5  | Contoh Dijkstra Graf (H) . . . . .  | 14 |
|      |   |    |
| 4.1  | Graf Peta . . . . .   | 17 |
| 4.2  | $A \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .               | 19 |
| 4.3  | $B \rightarrow A \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . . | 19 |
| 4.4  | $C \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .               | 20 |
| 4.5  | $D \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .                             | 20 |
| 4.6  | $E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .                             | 20 |
| 4.7  | $F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 21 |
| 4.8  | $G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 21 |
| 4.9  | $H \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 21 |
| 4.10 | $I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 22 |
| 4.11 | $J \rightarrow M$ . . . . .   | 22 |
| 4.12 | $K \rightarrow L \rightarrow M$ . . . . .   | 22 |
| 4.13 | $L \rightarrow M$ . . . . .   | 23 |
| 4.14 | $A \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .               | 23 |
| 4.15 | $B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . . | 24 |
| 4.16 | $C \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .               | 24 |
| 4.17 | $D \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .                             | 24 |
| 4.18 | $B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . . | 25 |
| 4.19 | $F \rightarrow G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 25 |
| 4.20 | $G \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 25 |
| 4.21 | $H \rightarrow I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 26 |
| 4.22 | $I \rightarrow K \rightarrow L \rightarrow M$ . . . . .   | 26 |

|      |                                 |    |
|------|---------------------------------|----|
| 4.23 | $J \rightarrow M$               | 26 |
| 4.24 | $K \rightarrow L \rightarrow M$ | 27 |
| 4.25 | $L \rightarrow M$               | 27 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Kata transportasi berasal dari bahasa latin *trans* yang berarti seberang atau sebelah lain dan kata *portarte* yang berarti mengangkut atau membawa. Transportasi adalah proses memindahkan barang ataupun orang menuju tempat tujuan. Menurut UU Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, transportasi adalah perpindahan orang dan/atau barang dari satu tempat ke tempat lain menggunakan kendaraan di ruang lalu lintas jalan. Perekonomian suatu wilayah sangat bergantung kepada transportasi. Menurut Morlok (1978), jaringan konektivitas yang sangat penting untuk perkembangan di bidang ekonomi dan sosial diciptakan oleh transportasi. Salah satu bentuk penting transportasi yang meningkatkan perkembangan di suatu wilayah adalah transportasi barang. Transportasi barang adalah transportasi yang melibatkan memindahkan bahan makanan, minuman, perabotan, dan benda-benda lainnya. Transportasi jenis ini biasa terjadi ketika seorang konsumen membeli barang dari penjual atau produsen.

Saat ini, teknologi semakin berkembang dan arus aliran barang semakin tinggi. Hal ini diakibatkan karena mudahnya pembelian barang yang berasal dari tempat yang jauh tanpa harus pergi secara pribadi ke penjual. Melalui media *online shop* yang sekarang sudah menjadi komponen penting dalam kehidupan sehari-hari, pembeli hanya perlu mengoperasikan prosedur pembelian melalui *handphone* masing-masing, kemudian menunggu barang yang dibeli untuk diantarkan oleh jasa pengiriman ke alamat tujuan.

Pada penelitian ini, akan dicari jarak dan rute terpendek diantara titik sumber yaitu pusat sortir yang ada di kota-kota pada pulau Jawa menuju titik tujuan yaitu pusat sortir yang ada di Bandar Lampung. Penelitian ini menggunakan metode *Shortest Path*.

## **1.2 Rumusan Masalah**

Rumusan masalah dalam penelitian ini adalah menganalisa dua algoritma yaitu Floyd Warshall dan Dijkstra untuk menentukan rute dan jarak terpendek dengan kasus beberapa titik sumber dengan satu titik tujuan.

## **1.3 Batasan Masalah**

Batasan masalah dalam penelitian ini adalah :

1. Titik sumber adalah pusat sortir JnT di kota besar dan sedang yang ada di pulau Jawa.
2. Kota yang dipilih sebagai titik sumber adalah kota yang memiliki pusat sortir JnT.
3. Titik tujuan adalah kota Bandar Lampung yang ada di Lampung.
4. Satuan yang digunakan adalah satuan jarak yaitu kilometer(Km).
5. Algoritma yang digunakan adalah algoritma Floyd-Warshall dan algoritma Dijkstra.

## **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah :

- (a) Mencari jarak dan rute terpendek dari titik sumber menuju titik tujuan.
- (b) Menganalisa penggunaan algoritma Floyd Warshall dan Dijkstra dalam menentukan rute terpendek dalam kasus banyak titik sumber dan satu titik tujuan.

## **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini adalah.

1. Menambah wawasan baru bagi pembaca.
2. Menentukan algoritma yang optimal untuk menentukan rute terpendek dalam kasus tertentu.
3. Sebagai referensi untuk jenis permasalahan yang serupa.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Transportasi**

Transportasi telah menjadi penopang perpindahan barang dan manusia dari satu tempat ke tempat lain, membentuk suatu jaringan vital yang menghubungkan masyarakat secara global. Melalui berbagai sudut pandang, para ahli telah memberikan definisi yang bervariasi mengenai konsep transportasi. Menurut Morlok (1978), transportasi adalah aktivitas pemindahan atau pengangkutan sesuatu dari satu lokasi ke lokasi lainnya, menciptakan jaringan konektivitas yang vital bagi perkembangan ekonomi dan sosial. Menurut UU Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, transportasi adalah perpindahan orang dan/atau barang dari satu tempat ke tempat lain menggunakan kendaraan di ruang lalu lintas jalan.

Dapat disimpulkan bahwa transportasi adalah suatu sistem atau proses yang memungkinkan pergerakan manusia dan barang untuk mencapai tujuan tertentu menggunakan kendaraan di ruang lalu lintas jalan. Definisi tersebut mencerminkan pentingnya transportasi dalam mendukung mobilitas manusia, pertumbuhan ekonomi, dan integrasi sosial di seluruh dunia.

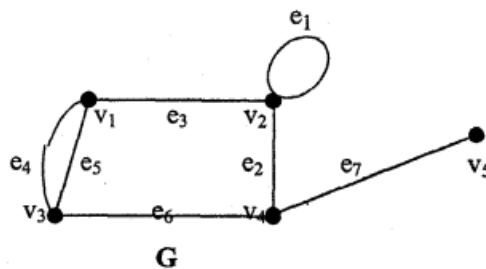
#### **2.2 Teori Graf**

Teori graf pertama kali dikenalkan pada tahun 1736 melalui tulisan Euler tentang upaya menyelesaikan permasalahan jembatan Königsberg yang terkenal di Eropa. Permasalahan ini membahas bagaimana cara melewati tujuh jembatan yang menghubungkan daratan yang dipisahkan oleh sungai Pregel dengan pulau kecil yang ada di tengah sungai tanpa melewati jembatan dua kali.

Pengembangan teori graf sangat intensif pada tiga puluh tahun terakhir. Banyak penelitian yang dilakukan, lusinan buku dan ribuan artikel yang telah ditulis dan diterbitkan baik itu murni maupun terapan (Buhaerah, 2022).

Graf ( $G$ ) didefinisikan sebagai himpunan objek yang terdiri dari titik ( $V$ ) dan sisi ( $E$ ) dengan  $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$  dan  $E(G) = \{e_1, e_2, e_3, \dots, e_n\}$ . Dimana  $V(G)$  adalah himpunan titik ( $V$ ) yang tak kosong sedangkan  $E(G)$  adalah himpunan  $E$  dimana setiap sisi  $e_n$  diidentifikasi dengan pasangan titik yang tak berurutan sebagai ujungnya. Artinya suatu sisi bisa memiliki titik yang sama di ujungnya, sisi ini biasa disebut sebagai gelang (Deo, 2016).

Cara merepresentasikan graf yang paling umum adalah dengan menggunakan diagram. Dalam diagram tersebut titik dalam graf akan dihubungkan oleh sisi yang berbentuk seperti garis lurus ataupun kurva, contohnya seperti pada gambar di bawah ini.



**Gambar 2.1 Contoh Graf ( $G$ )**

Beberapa teminologi pada graf adalah : gelang, sisi rangkap, derajat, bertetangga, bersisian, graf sederhana, graf tidak sederhana, graf berarah, graf tak berarah, *walk*, *trail*, *path*, *cycle* dan graf berbobot.

Dari gambar di atas dapat dilihat bahwa setiap titik dihubungkan oleh sisi. Dalam graf, sisi yang kedua ujungnya berada pada titik yang sama disebut sebagai gelang (*loop*) seperti  $e_1$  pada gambar. Dalam graf juga sepasang titik bisa dihubungkan oleh dua atau lebih sisi, hal ini disebut sebagai sisi rangkap.

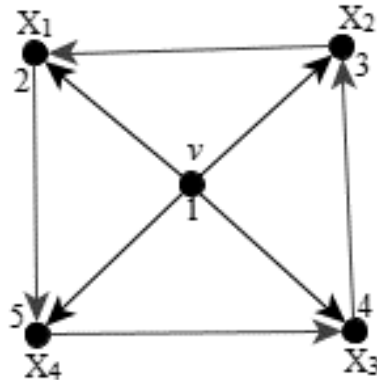
Pada Gambar 2.1 dapat dilihat bahwa dalam satu titik bisa dihubungkan dengan beberapa sisi. Jumlah atau banyaknya sisi yang menempel pada suatu titik disebut sebagai derajat (*degree*). Contohnya titik  $v_1$  pada Gambar 2.1 memiliki derajat tiga.

Dua titik pada graf dikatakan bertetangga jika keduanya dihubungkan oleh sisi. Contohnya titik  $v_2$  bertetangga dengan titik  $v_1$ . Sisi yang menghubungkan dua titik tersebut dikatakan bersisian dengan titik  $v_2$  dan titik  $v_1$ .

Graf dibagi menjadi dua berdasarkan ada tidaknya sisi rangkap atau gelang pada graf. Yaitu, graf sederhana dan graf tidak sederhana. Graf sederhana

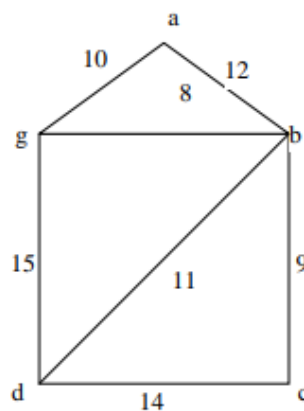
adalah graf yang tidak memiliki sisi rangkap dan gelang. Graf tidak sederhana adalah graf yang memiliki sisi rangkap atau gelang.

Graf juga dibagi menjadi dua berdasarkan orientasi arah pada sisi graf. Di antaranya, graf berarah dan graf tak berarah. Graf berarah adalah graf yang setiap sisinya memiliki orientasi arah. Sedangkan graf tak berarah adalah graf yang sisinya tidak memiliki orientasi arah.



**Gambar 2.2 Contoh Graf Berarah**

Suatu *walk* (jalan) adalah suatu urutan titik-titik dan sisi-sisi yang bergantian, dimana setiap sisi bersisian dengan titik terdekat dan titik harus bertetangga dengan titik sebelumnya, *walk* (jalan) diawali dan diakhiri pada suatu titik. *Walk* yang setiap sisinya hanya dilalui sekali maka disebut sebagai *trail*. *Trail* yang setiap titiknya hanya dilalui sekali disebut *path* (rute). Rute juga bisa didefinisikan sebagai barisan sisi yang menghubungkan titik yang berbeda. *Path* (rute) yang memiliki titik awal dan titik akhir sama disebut sebagai *cycle* (siklus). Graf berbobot adalah graf yang setiap sisinya diberikan bobot atau harga, seperti pada gambar berikut ini.



**Gambar 2.3 Contoh Graf berbobot**

### 2.3 Shortest Path Problem

*Shortest Path Problem* adalah algoritma yang digunakan dalam menganalisis pemmasalahan jaringan (*network*), fungsi utama algoritma ini adalah untuk menentukan jalur/rute terpendek antar titik awal (sumber) dengan titik tujuan. Atau bisa juga diartikan sebagai cara untuk menentukan rute antara dua atau lebih titik dalam graf berbobot dimana gabungan sisi yang dilalui memiliki nilai paling kecil (Wulandari, 2022).

#### 2.3.1 Algoritma Floyd-Warshall

Algoritma yang bisa digunakan untuk mencari rute terpendek dari semua titik yang ada pada graf adalah algoritma Floyd-Warshall. Dengan menggunakan algoritma ini, akan diketahui rute terpendek dari suatu titik menuju semua titik pada graf. Algoritma ini bisa digunakan untuk mencari rute terpendek pada graf berbobot positif atau negatif (dengan syarat tidak terdapat siklus negatif pada graf). Algoritma ini membandingkan setiap pasangan titik untuk mencari rute terpendek dengan nilai akurasi algoritma adalah 100% (Gunawan, 2021).

Langkah-langkah yang diperlukan untuk menyelesaikan suatu permasalahan *shortest path* dengan algoritma Floyd-Warshall adalah :

(a) Graf diberikan keterangan :

$i$  = titik awal

$j$  = titik tujuan

$e_{ij}$  = sisi yang menghubungkan  $i$  dan  $j$

$k$  = titik perantara

$D_{ij}$  = jarak antara  $i$  dan  $j$

(b) Merubah graf menjadi berbobot menjadi matriks dengan bobot :

Jika  $i = j$

$D_{ij} = 0$

Jika  $i \neq j$  dan  $e_{ij} \in E$

$D_{ij} = D[i, j]$

Jika  $i \neq j$  dan  $e_{ij} \notin E$

$D_{ij} = \infty$

Dengan hasil atau *output* bentuk matriksnya adalah  $n \times n$  dengan jarak  $D_{ij}$ .

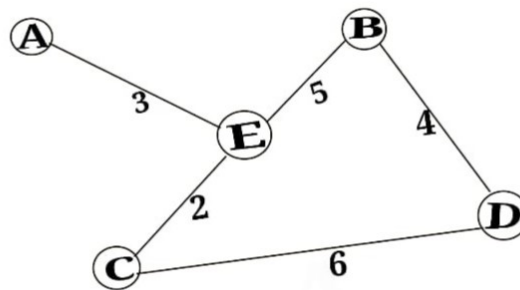
(c) Buat matriks  $n \times n$  dengan  $n$  adalah jumlah titik yang ada pada graf.

Isi matriks dengan bobot yang ada pada graf, jika tidak ada sisi yang menghubungkan secara langsung antara titik maka bobot ditulis  $\infty$ .

- (d) Gunakan matriks sebagai representasi yang akan digunakan untuk melakukan perhitungan. Rumus untuk mencari jarak terpendeknya adalah  $D[i,j] = \min(D[i,j], D[k,j]+D[i,k])$ .
- (e) Pilih salah satu titik sebagai titik perantara, misalkan dipilih titik A maka cara mencari jarak dari titik B menuju titik D adalah  $D[i,j] = \min(D[B,D], D[A,B]+D[D,A])$ .
- (f) Ulangi terus menerus sampai semua jarak terpendek dengan perantara titik A ditemukan. Kemudian didapatkan iterasi 1.
- (g) Setelah didapatkan matriks iterasi 1, pilih titik perantara berikutnya yang belum digunakan kemudian ulangi perhitungan tersebut sampai didapatkan matriks iterasi n (Bari, 2018).

Contoh algoritma Floyd-Warshall adalah :

Terdapat sebuah graf berbobot (G) sebagai berikut :



Gambar 2.4 Contoh Floyd-Warshall Graf (G)

Langkah-langkah untuk mencari jarak terpendek dari graf tersebut adalah.

- (a) Rubah graf ke dalam bentuk matriks.

$$\begin{bmatrix} 0 & \infty & \infty & \infty & 3 \\ \infty & 0 & \infty & 4 & 5 \\ \infty & \infty & 0 & 6 & 2 \\ \infty & 4 & 6 & 0 & \infty \\ 3 & 5 & 2 & \infty & 0 \end{bmatrix} \quad (2.3.1)$$

(b) Iterasi 1

Pilih titik A sebagai perantara.

$$\begin{bmatrix} 0 & \infty & \infty & \infty & 3 \\ \infty & 0 & \infty & 4 & 5 \\ \infty & \infty & 0 & 6 & 2 \\ \infty & 4 & 6 & 0 & \infty \\ 3 & 5 & 2 & \infty & 0 \end{bmatrix}$$

$$\min (D[B, C], D[A, B] + D[A, C])$$

$$\min (\infty, \infty + \infty)$$

$$\min (D[B, D], D[A, B] + D[A, D])$$

$$\min (4, \infty + \infty)$$

$$\min (D[B, E], D[A, B] + D[A, E])$$

$$\min (5, \infty + 3)$$

$$\min (D[C, D], D[A, C] + D[A, D])$$

$$\min (6, \infty + \infty)$$

$$\min (D[C, E], D[A, C] + D[A, E])$$

$$\min (2, \infty + 3)$$

$$\min (D[D, E], D[A, D] + D[A, E])$$

$$\min (\infty, \infty + 3)$$

(c) Iterasi 2

Pilih titik B sebagai perantara.

$$\begin{bmatrix} 0 & \infty & \infty & \infty & 3 \\ \infty & 0 & \infty & 4 & 5 \\ \infty & \infty & 0 & 6 & 2 \\ \infty & 4 & 6 & 0 & 9 \\ 3 & 5 & 2 & 9 & 0 \end{bmatrix}$$

$$\min (D[A, C], D[A, B] + D[B, C])$$

$$\min (\infty, \infty + \infty)$$

$$\min (D[A, D], D[A, B] + D[B, D])$$

$$\min (\infty, \infty + 4)$$

$$\min (D[A, E], D[A, B] + D[B, E])$$

$$\min (3, \infty + 5)$$

$$\min (D[C, D], D[B, C] + D[B, D])$$

$$\min (6, \infty + 4)$$

$$\min (D[C, E], D[B, C] + D[B, E])$$

$$\min (2, 5 + \infty)$$

$$\min (D[D, E], D[B, D] + D[B, E])$$

$$\min (\infty, 4 + 5)$$

(d) Iterasi 3

Pilih C sebagai titik perantara.

$$\begin{bmatrix} 0 & \infty & \infty & \infty & 3 \\ \infty & 0 & \infty & 4 & 5 \\ \infty & \infty & 0 & 6 & 2 \\ \infty & 4 & 6 & 0 & 8 \\ 3 & 5 & 2 & 8 & 0 \end{bmatrix}$$

$$\min (D[A, B], D[A, C] + D[B, C])$$

$$\min (\infty, \infty + \infty)$$

$$\min (D[A, D], D[A, C] + D[C, D])$$

$$\min (\infty, \infty + 6)$$

$$\min (D[A, E], D[A, C] + D[C, E])$$

$$\min (3, \infty + 2)$$

$$\min (D[B, D], D[B, C] + D[C, D])$$

$$\min (4, \infty + 6)$$

$$\min (D[B, E], D[B, C] + D[C, E])$$

$$\min (5, \infty + 2)$$

$$\min (D[D, E], D[C, D] + D[C, E])$$

$$\min (9, 6 + 2)$$

(e) Iterasi 4

Pilih D sebagai titik perantara.

$$\begin{bmatrix} 0 & \infty & \infty & \infty & 3 \\ \infty & 0 & 10 & 4 & 5 \\ \infty & 10 & 0 & 6 & 2 \\ \infty & 4 & 6 & 0 & 8 \\ 3 & 5 & 2 & 8 & 0 \end{bmatrix}$$

$$\min (D[A, B], D[A, D] + D[B, D])$$

$$\min (\infty, \infty + 4)$$

$$\min (D[A, C], D[A, D] + D[C, D])$$

$$\min (\infty, \infty + 6)$$

$$\min (D[A, E], D[A, D] + D[D, E])$$

$$\min (3, \infty + 8)$$

$$\min (D[B, C], D[B, D] + D[C, D])$$

$$\min (\infty, 4 + 6)$$

$$\min (D[B, E], D[B, D] + D[D, E])$$

$$\min (5, 4 + 8)$$

$$\min (D[C, E], D[C, D] + D[D, E])$$

$$\min (2, 6 + 8)$$

(f) Iterasi 5

Pilih E sebagai titik perantara.

$$\begin{bmatrix} 0 & 8 & 5 & 11 & 3 \\ 8 & 0 & 7 & 4 & 5 \\ 5 & 7 & 0 & 6 & 2 \\ 11 & 4 & 6 & 0 & 8 \\ 3 & 5 & 2 & 8 & 0 \end{bmatrix}$$

$$\min (D[A, B], D[A, E] + D[B, E])$$

$$\min (\infty, 3 + 5)$$

$$\min (D[A, C], D[A, E] + D[C, E])$$

$$\min (\infty, 3 + 2)$$

$$\min (D[A, D], D[A, E] + D[D, E])$$

$$\min (\infty, 3 + 8)$$

$$\min (D[B, C], D[B, E] + D[C, E])$$

$$\min (10, 5 + 2)$$

$$\min (D[B, D], D[B, E] + D[D, E])$$

$$\min (4, 5 + 8)$$

$$\min (D[C, D], D[C, E] + D[D, E])$$

$$\min (6, 2 + 8)$$

(g) Setelah iterasi ke 5 atau iterasi ke n, didapatkan jarak-jarak terpendek setiap di antara dua titik.

### 2.3.2 Algoritma Dijkstra

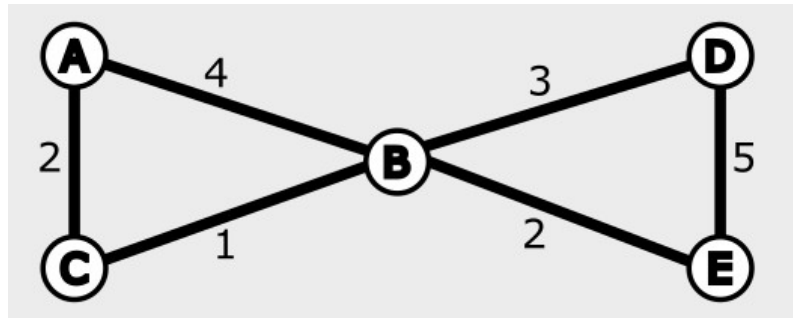
Algoritma yang biasanya digunakan untuk mencari rute terpendek dari suatu titik sumber menuju titik tujuan adalah algoritma Dijkstra. Nama Dijkstra diambil dari penemu algoritma ini yang bernama Edsger Dijkstra. Algoritma ini termasuk ke dalam algoritma greedy yang dimana rute atau jalur yang dipilih adalah rute dengan gabungan sisi yang memiliki bobot minimum yang menghubungkan setiap titik yang dilalui untuk menghubungkan titik sumber ke titik tujuan. Algoritma Dijkstra bekerja dengan cara mencari nilai yang paling minimal pada graf berbobot, sehingga graf dengan dua atau lebih titik akan diketahui rute terpendeknya (Ahadi, 2022).

Langkah-langkah yang harus dilakukan untuk menggunakan algoritma Dijkstra adalah sebagai berikut :

- (a) Memberikan bobot pada setiap sisi diantara titik, kemudian pilih dan berikan nilai (0) pada titik sumber dan nilai ( $\infty$ ) pada titik lainnya.
- (b) Setelah menentukan titik sumber, tentukan titik tujuan atau titik akhir. Perhatikan rute atau jalur yang bisa menghubungkan titik awal dan titik tujuan.
- (c) Kemudian pertimbangkan titik yang belum diberikan nilai, berdekatan dan terhubung langsung dengan titik awal dan pilih sisi dengan bobot minimal. Misalkan titik awal (A) terhubung langsung pada titik B dan titik C dengan bobot ke titik B adalah 6 dan bobot ke titik C adalah 4, maka pilih titik C yang membuat nilai titik C menjadi 4. Selanjutnya titik C terhubung ke titik D dan titik E dengan bobot ke titik D adalah 4 dan bobot ke titik E adalah 5, maka pilih titik D yang membuat gabungan bobot dari titik A menuju titik D menjadi  $4+4 = 8$ , maka nilai titik D adalah 8. Ulangi proses tersebut sampai didapatkan rute dari titik awal menuju titik tujuan.
- (d) Setelah rute tersebut didapatkan, ulangi proses tersebut dengan titik terdekat yang belum diberikan nilai. Nilai yang disimpan pada titik adalah bobot terakhir dengan nilai paling minimal.
- (e) Ketika semua titik yang dapat dilalui memiliki nilai, maka untuk menentukan rute terpendek adalah dengan memilih rute dengan bobot total atau nilai paling minimal.

Contoh dari algoritma Dijkstra adalah :

Cari jarak terpendek dari titik A menuju titik D pada graf berbobot (H) sebagai berikut :



Gambar 2.5 Contoh Dijkstra Graf (H)

Langkah-langkah untuk mencari rute terpendek dari graf tersebut adalah.

- (a) Titik yang terhubung langsung dengan Titik A adalah titik B dengan bobot 4 dan titik C dengan bobot 2. Maka pilih titik C.
- (b) Titik C terhubung langsung dengan titik B dengan bobot 1, maka jarak titik A menuju titik B melalui rute  $A \rightarrow C \rightarrow B$  adalah 3. Sedangkan titik A langsung menuju B berjarak 4, maka rubah nilai B menjadi yang paling minimal yaitu 3.
- (c) Titik B terhubung langsung dengan titik D dengan bobot 3 dan titik E dengan bobot 2. Pilih titik E sehingga didapat nilai 5.
- (d) Titik E terhubung dengan titik D dengan bobot 5, maka didapat nilai dari rute  $A \rightarrow C \rightarrow B \rightarrow E \rightarrow D$  adalah 10. Sedangkan jika dari titik B langsung menuju titik D memiliki bobot 3, maka rute  $A \rightarrow C \rightarrow B \rightarrow D$  memiliki nilai 6.
- (e) Sehingga dipilih rute  $A \rightarrow C \rightarrow B \rightarrow D$  yang memiliki nilai lebih sedikit dari rute  $A \rightarrow C \rightarrow B \rightarrow E \rightarrow D$ .

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Penelitian ini dilakukan pada semester ganjil tahun ajaran 2025/2026 di Jurusan Matematika Fakultas Ilmu Pengetahuan Alam dan Matematika Universitas Lampung yang beralamat di Jalan Prof. Dr. Ir. Soemantri Brojonegoro, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Lampung.

#### **3.2 Langkah Penelitian**

Langkah penelitian merupakan tahapan-tahapan yang dilakukan peneliti dalam menjalankan penelitian ini, berikut tahapan-tahapan yang dilakukan peneliti :

##### **3.2.1 Pengumpulan dan Pemahaman Materi Terkait**

Pada langkah ini, peneliti akan mengumpulkan materi atau studi kasus yang berkaitan untuk memperkuat pemahaman peneliti dan mempermudah jalannya penelitian untuk kedepannya nanti.

##### **3.2.2 Pengumpulan Data**

Peneliti mencari data melalui media-media jurnal, artikel, dan web resmi yang berkaitan.

##### **3.2.3 Membuat Graf**

Setelah didapatkan data yang dirasa cukup, peneliti akan membuat graf dari data-data tersebut untuk meringkas dan mempermudah pengolahan data. Pada tahap ini, peneliti menggunakan alat bantuan berupa program atau aplikasi untuk mendesain dan menggambar graf.

### **3.2.4 Perhitungan Menggunakan Program**

Peneliti akan menggunakan alat bantu berupa program Python untuk mempermudah dan mempersingkat perhitungan yang dibutuhkan untuk menyelesaikan penelitian ini.

### **3.2.5 Analisa**

Pada tahap ini, peneliti akan menganalisa hasil perhitungan kedua algoritma.

### **3.2.6 Menarik Kesimpulan**

Tahap ini adalah penutup dalam proses penelitian ini, yaitu peneliti akan menarik kesimpulan dari hasil analisa yang telah dilakukan pada tahap-tahap sebelumnya.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setiap algoritma memiliki keunggulan dan kekurangannya masing-masing. Algoritma Floyd Warshall dapat digunakan untuk mencari *shortest path* hanya dengan melakukan satu kali *input* dan *running* di dalam Python. Pada algoritma ini setiap titik pada graf akan diketahui jarak terpendek menuju titik lainnya yang ada pada graf dan hasilnya lebih akurat daripada algoritma Dijkstra. Karena keterbatasan Python, iterasi yang ditampilkan pada bagian *output* hanya iterasi terakhir atau iterasi ke-n. Sehingga tidak dapat diketahui rute yang dilalui oleh program untuk mendapatkan jarak terpendeknya. Hal ini disebabkan karena kurangnya data iterasi ke-1 sampai iterasi ke-(n-1). Karena itu, dibutuhkan bantuan dari media atau algoritma lain untuk mencari data pelengkap.

Algoritma Dijkstra berbeda dengan algoritma Floyd Warshall, setiap sepasang titik yang ada pada graf dibutuhkan satu kali *input* dan *running* di dalam Python. Pada algoritma ini, tidak didapatkan secara langsung seluruh jarak terpendek di antara dua titik dalam graf. Dibutuhkan perhitungan tersendiri untuk setiap sepasang titik sumber dan titik tujuan. Sebagai gantinya, bagian *output* menampilkan jarak terpendek, rute yang ditempuh, dan peta visual. Sehingga hasil lebih lengkap dan mudah dibaca dari pada algoritma Floyd Warshall.

## 5.2 Saran

Disarankan praktisi menggunakan program lain untuk mengatasi keterbatasan program Python yang tidak menampilkan data iterasi pertama sampai iterasi ke-n. Sehingga dalam proses penentuan rute yang diambil pada algoritma Floyd Warshall, dapat digunakan data iterasi-iterasi tersebut untuk mencari rutanya. Untuk penelitian selanjutnya bisa digunakan data yang berbeda dan metode *shortest path* lainnya, seperti data yang memiliki bobot negatif dengan menggunakan algoritma Floyd Warshall dan Bellman Ford yang dapat menyelesaikan bobot negatif.

## DAFTAR PUSTAKA

- Ahadi, I., Mimi, N.H., Popy, P.D.D., & Muchammad, F. (2022). Penerapan Algoritma Dijkstra untuk Mencari Rute Terpendek pada Pengiriman Produk Wafer di PT. XYZ. *Jurnal Manajemen Teknologi dan Teknik Industri Universitas Kadiri*, 4(1).
- Bari, A. (2018, 16 Februari). *4.2 All Pairs Shortest Path (Floyd-Warshall)-Dynamic Programming* [Video]. YouTube. <https://www.youtube.com/watch?v=oNI0rf2P9gE>
- Buhaerah, Zulfiqar, B., & Herlan, S. (2022). *Teori Graf dan Aplikasinya*. Makassar : LSQ.
- Bunaen, M.C., Hanna, P., & Yosefina, F.R. (2022). Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya Ke Tempat Bersejarah. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 4(1).
- Deo, N. (2016). *Graph Theory With Applications to Engineering and Computer Science*. New York : Dover Publication.
- Gunawan, & Wresti, A. (2021). Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall Penentuan Jalur Lintasan Terpendek Stasiun Tegal Menuju Hotel. *Jurnal BATIRSI*, 4(2).
- Hamdi, S., & Prihandoko. (2018). Analisis Algoritma Dijkstra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota). *Jurnal ENERGY*, 8(1).
- Indonesia. *Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan*. Lembaran Negara RI Tahun 2009 No 96. Tambahan Lembaran Negara RI No 5025. Sekretariat Negara. Jakarta.
- Johnsonbough, R. (2018). *Discrete Mathematics Eighth Edition*. USA : Pearson Education, Inc.
- Meflinda, A., & Mahyarni. (2011). *Operation Reserch (Riset Operasi)*. Riau : Unri Press.

Morlok, Edward K. 1988. *Pengantar Teknik dan Perencanaan Transportasi*. Jakarta: Erlangga.

Wulandari, I.K., & Pristi, S. (2022). Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan. *JURNAL ILMIAH SISTEM INFORMASI*, 1(1).