

**PENERAPAN ALGORITMA FLOYD-WARSHALL PADA
PENENTUAN JALUR TERPENDEK EVAKUASI KORBAN
KECELAKAAN DI BOYOLALI JAWA TENGAH**

SKRIPSI

**OLEH:
FITRI NOFITA SARI
NIM. 19610004**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**PENERAPAN ALGORITMA FLOYD-WARSHALL PADA
PENENTUAN JALUR TERPENDEK EVAKUASI KORBAN
KECELAKAAN DI BOYOLALI JAWA TENGAH**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
FITRI NOFITA SARI
NIM. 19610004**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**PENERAPAN ALGORITMA FLOYD-WARSHALL PADA
PENENTUAN JALUR TERPENDEK EVAKUASI KORBAN
KECELAKAAN DI BOYOLALI JAWA TENGAH**

SKRIPSI

**Oleh
Fitri Nofita Sari
NIM. 19610004**

Telah Diperiksa dan Disetujui Untuk Diuji
Malang, 31 Mei 2023

Dosen Pembimbing I



Juhari, M.Si.

NIDT. 19840209 20160801 1 055

Dosen Pembimbing II



Erna Herawati, M.Pd.

NIDT. 1976072320180201 2 222

Mengetahui,
Ketua Program Studi Matematika



Dr. Elly Susanti, M.Sc.

NIP. 19741129 200012 2 005

**PENERAPAN ALGORITMA FLOYD-WARSHALL PADA
PENENTUAN JALUR TERPENDEK EVAKUASI KORBAN
KECELAKAAN DI BOYOLALI JAWA TENGAH**

SKRIPSI

**Oleh
Fitri Nofita Sari
NIM. 19610004**

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 7 Juni 2023

Ketua Penguji : Dr. Elly Susanti, M.Sc.

Anggota Penguji I : Muhammad Khudzaifah, M.Si.

Anggota Penguji II : Juhari, M.Si.

Anggota Penguji III : Erna Herawati, M.Pd.



Mengetahui,
Ketua Program Studi Matematika




Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini:

Nama : Fitri Nofita Sari
NIM : 19610004
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Penerapan Algoritma Floyd-Warshall Pada Penentuan Jalur Terpendek Evakuasi Korban Kecelakaan Di Boyolali Tengah

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 7 Juni 2023
Yang membuat pernyataan,



Fitri Nofita Sari
NIM. 19610004

MOTO

“Jangan takut jatuh, karena yang tidak pernah memanjatlah yang tidak pernah jatuh. Yang takut gagal, karena yang tidak pernah gagal hanyalah orang-orang yang tidak pernah melangkah. Jangan takut salah, karena dengan kesalahan yang pertama kita dapat mendapat pengetahuan untuk mencari jalan keluar yang benar pada langkah kedua.”

(Buya Hamka)

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Bapak Suparman, Ibu Suwarni, Kakak Ista Khanudhin, dan Seluruh keluarga tersayang yang selalu memberikan dukungan penulis berupa moral ataupun material. Serta sahabat-sahabat penulis yang senantiasa memberikan semangat dan do'a. Serta Mamas yang senantiasa memberikan semangat dan menemani penulis dalam setiap keadaan.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Warakatuh

Puji syukur kehadiran Allah Subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayah sehingga penulis mampu menulis skripsi yang berjudul “Penerapan Algoritma Floyd-Warshall Pada Penentuan Jalur Terpendek Evakuasi Korban Kecelakaan Di Boyolali Jawa Tengah”. Shalawat serta salam kepada baginda Nabi Muhammad Shalallahu Alaihi Wassalam yang telah menunjukkan dan membimbing kita dari zaman jahiliah menuju zaman yang terang benderang.

Tidak lupa pula dukungan baik secara materil dan nonmateril yang diberikan kepada penulis dari berbagai pihak. Oleh karena itu, izinkan penulis mengucapkan rasa terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Elly Susanti, M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Juhari, M.Si, selaku dosen pembimbing I yang banyak memberikan arahan, bimbingan, saran dan motivasi dalam penulisan penyelesaian skripsi.
5. Erna Herawati, M.Pd, selaku dosen pembimbing II yang banyak memberikan arahan, bimbingan, saran dan motivasi dalam penulisan penyelesaian skripsi.
6. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim
7. Bapak Suparman, dan Ibu Suwarni selaku orang tua, Ibu Parmi dan seluruh keluarga yang tiada henti memberikan dukungan dan doa kepada penulis.
8. Mas Moh. Erfan Fahrudin yang senantiasa membersamai dan memberikan semangat kepada penulis dan menemani suka maupun duka.
9. Habiba Amalia, Ni'matul Azizah, Rina, Eliya Nur, Aniq, Gita dan teman-teman kontrakan yang senantiasa memberikan semangat, menghibur, dan bantuan dalam bentuk moril maupun yang lainnya.

10. Seluruh teman dan sahabat serta mahasiswa angkatan 2019 yang memeberikan semangat, cerita dan pengalam yang berkesan selama menempuh perkuliahan.
11. Seluruh pihak yang tidak bisa penulis sebut satu-persatu atas segala dukungan dan bantuannya dalam proses penyelesaian skripsi ini.

Semoga Allah senantiasa membalas perbuatan baik mereka. Semoga skripsi ini dapat memeberikan manfaat khususnya bagi penulis dan pembaca, *aamiin*.

Wa'alaikumussalam Warahmatullahi Wabarakatuh.

Malang, 7 Juni 2023

Penulis

DAFTAR ISI

| | |
|---|-------------|
| HALAMAN JUDUL | i |
| HALAMAN PENGAJUAN | ii |
| HALAMAN PERSETUJUAN | iii |
| HALAMAN PENGESAHAN..... | iv |
| PERNYATAAN KEASLIAN TULISAN | v |
| MOTO..... | vi |
| PERSEMBAHAN..... | vii |
| KATA PENGANTAR..... | viii |
| DAFTAR ISI..... | x |
| DAFTAR TABEL | xii |
| DAFTAR GAMBAR..... | xiii |
| DAFTAR LAMPIRAN | xiv |
| ABSTRAK | xv |
| ABSTRACT | xvi |
| مستخلص البحث | xvii |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah | 7 |
| 1.3 Tujuan Penelitian..... | 7 |
| 1.4 Manfaat Penelitian..... | 7 |
| 1.5 Batasan Masalah..... | 7 |
| BAB II KAJIAN TEORI | 9 |
| 2.1 Teori Pendukung | 9 |
| 2.1.1 Definisi Graf..... | 9 |
| 2.1.2 Matriks..... | 12 |
| 2.1.3 Algoritma | 18 |
| 2.1.4 Algoritma Floyd Warshall..... | 18 |
| 2.2 Kajian Integrasi dengan Al-Qur'an dan Hadits | 21 |
| 2.3 Kajian Topik Dengan Teori Pendukung..... | 22 |
| BAB III METODE PENELITIAN | 24 |
| 3.1 Jenis Penelitian | 24 |
| 3.2 Tahapan Penelitian | 25 |
| 3.3 Flowchart | 27 |
| BAB IV PEMBAHASAN..... | 28 |
| 4.1 Deskripsi Data | 28 |
| 4.2 Pengolahan Data..... | 30 |
| 4.3 Perhitungan Lintasan Terpendek..... | 31 |
| 4.4 Penentuan Rute Terpendek..... | 35 |
| 4.5 Integrasi Al-Quran dan Hadits Terkait Kecelakaan di Boyolali..... | 40 |
| BAB V PENUTUP..... | 43 |
| 5.1 Kesimpulan..... | 43 |
| 5.2 Saran..... | 43 |
| DAFTAR PUSTAKA | 44 |
| LAMPIRAN..... | 46 |
| RIWAYAT HIDUP | 58 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Perbedaan Graf Secara Ringkas | 12 |
| Tabel 2.2 Tabel Contoh Graf Ketetanggan | 13 |
| Tabel 4.1 Obyek Penelitian Lokasi Ruas Jalan Rawan Kecelakaan Boyolali . | 27 |
| Tabel 4.2 Daftar Titik dari Graf Wilayah Penelitian..... | 30 |

DAFTAR GAMBAR

| | | |
|------------|---|----|
| Gambar 2.1 | Graf G..... | 9 |
| Gambar 2.2 | Graf Sederhana..... | 10 |
| Gambar 2.3 | Graf Tak-Sederhana | 10 |
| Gambar 2.4 | Graf Tak-Berarah | 11 |
| Gambar 2.5 | Graf Berarah..... | 12 |
| Gambar 2.6 | Contoh Matriks Ketetanggan | 13 |
| Gambar 2.7 | Prinsip Algoritma | 15 |
| Gambar 3.1 | Flowchart Algoritma Floyd-Warshall | 26 |
| Gambar 4.1 | Dokumentasi Konfirmasi Data di Polres Boyolali..... | 28 |
| Gambar 4.2 | Titik Lokasi Kecelakaan..... | 29 |
| Gambar 4.3 | Lokasi Sebaran Rumah Sakit | 30 |
| Gambar 4.4 | Gambar Wilayah Penelitian | 31 |
| Gambar 4.5 | Hasil Percobaan Rute A ke 1..... | 40 |

DAFTAR LAMPIRAN

| | | |
|------------|---|----|
| Lampiran 1 | Script Program Algoritma Floyd-Warshall dengan Matlab | 46 |
| Lampiran 2 | Tabel Jarak Antar Titik Sebelum Dilakukan Perhitungan..... | 48 |
| Lampiran 3 | Matriks Iterasi $k = 0$ | 49 |
| Lampiran 4 | Matriks Iterasi $k = 1$ | 50 |
| Lampiran 5 | Matriks Iterasi $k = 2$ | 51 |
| Lampiran 6 | Matriks Iterasi $k = 3$ | 52 |
| Lampiran 7 | Matriks Iterasi $k = 23$ | 53 |
| Lampiran 8 | Jarak Antar Titik Setelah Dilakukan Perhitungan | 54 |
| Lampiran 9 | Uji Manual | 55 |

ABSTRAK

Sari, Fitri Nofita. 2023. **Penerapan Algoritma Floyd-Warshall Pada Penentuan Jalur Terpendek Evakuasi Korban Kecelakaan Di Boyolali JawaTengah**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Juhari. M.Si, (II) Erna Herawati, M.Pd.

Kata Kunci: Rute Terpendek, Graf, Algoritma Floyd-Warshall, Titik.

Algoritma Floyd-Warshall adalah salah satu algoritma yang dapat digunakan untuk menyelesaikan persoalan rute terpendek dan paling mudah untuk diterapkan karena dapat menemukan semua rute terpendek antara setiap pasangan posisi yang dimungkinkan dan merupakan bagian dari program dinamis yang sangat efisien dalam memecahkan persoalan rute yang optimal. Algoritma Floyd-Warshall bekerja dengan cara membandingkan setiap kemungkinan jalur pada graf untuk setiap pasangan simpul dan melakukan pengecekan pada kombinasi simpul yang dihasilkan. Masalah rute terpendek dalam kehidupan sehari-hari adalah evakuasi korban kecelakaan lalu lintas di Boyolali, Jawa Tengah. Agar evakuasi lebih efektif, diperlukan rute terdekat ke rumah sakit terdekat. Berdasarkan hasil penelitian yang telah diiterasi sebanyak $n = k = 23$, disimpulkan bahwa penelitian menghasilkan matriks referensi berupa lintasan terpendek yang digunakan untuk menentukan rute terpendek menuju rumah sakit terdekat di Boyolali, Jawa Tengah. Rute ini dapat digunakan oleh masyarakat Boyolali dan sekitarnya serta instansi terkait mengenai rute terdekat dari jalan rawan kecelakaan rumah sakit.

ABSTRACT

Sari, Fitri Nofita. 2023. **On The Application of Floyd-Warshall Algorithm to Determination the Route of Accident Evacuation in Boyolali Central Java.** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor: (I) Juhari. M.Si, (II) Erna Herawati, M.Pd.

Keywords: Shortest Route, Graph, Floyd-Warshall Algorithm, Period.

The Floyd-Warshall algorithm is one of the algorithms that can be used to solve the shortest route problem and is the easiest to apply because it can find all the shortest routes between each possible pair of positions and is part of a dynamic program that is very efficient in solving optimal route problems. The Floyd-Warshall algorithm works by comparing each possible path on the graph for each node pair and checking the resulting node combination. The problem of the shortest route in daily life is the evacuation of victims of traffic accidents in Boyolali, Central Java. For evacuation to be more effective, a nearby route to the nearest hospital is needed. Based on the results of research that has been iterated as much as $n = k = 23$, it was concluded that the study produced a reference matrix in the form of the shortest trajectory used to determine the shortest route to the nearest hospital in Boyolali, Central Java. This route can be used by the people of Boyolali and surrounding areas as well as related agencies regarding the nearest route from the hospital accident-prone road towards.

مستخلص البحث

ساري، فيطري نوفيتا. ٢٠٢٣. تطبق خوارزمية (Floyd-Warshall) في تحديد المسار أقصر إجلاء لضحاي أحوادثفي بويولالي جاوي الوسط. الحث لعلمي قسم الرياضات، كلية العلوم والتكنولوجيا مولانا مالك إبراهيم الإسلامية الحكيمة مالانج. المشرف: (١) جوهاي، الماجستير (٢) إيرناهيراواقي، الماجستير

الكلمة املفتاحية: تطبق (Floyd-Warshall) أقصر الطريق ، رسم البياني ، الخوارزمية

خوارزمية (Floyd-Warshall) هي إحدى الخوارزميات التي يمكن استخدامها لحل مشكلة أقصر طريق وهي الأسهل في التطبيق لأنها يمكن أن تجد جميع أقصر الطرق بين كل زوج ممكن من المواضع وهي جزء من برنامج ديناميكي فعال للغاية في حل مشاكل المسار الأمثل. تعمل خوارزمية (Floyd-Warshall) من خلال مقارنة كل مسار ممكن على الرسم البياني لكل زوج من العقد والتحقق من . مشكلة أقصر طريق في الحياة اليومية هي إجلاء ضحايا حوادث المرور في بويولالي ، جاوة الوسطى. لكي يكون الإخلاء أكثر فعالية ، هناك حاجة إلى طريق قريب إلى أقرب مستشفى. استنادا إلى نتائج البحث التي تم تكرارها بقدر $(n = k = 23)$ ، تم استنتاج أن الدراسة أنتجت مصفوفة مرجعية في شكل أقصر مسار يستخدم لتحديد أقصر طريق إلى أقرب مستشفى في (Boyolali) ، جاوة الوسطى. يمكن استخدام هذا الطريق من قبل سكان بويولالي والمناطق المحيطة بها وكذلك الوكالات ذات الصلة فيما يتعلق بأقرب طريق من الطريق المعرض للحوادث في المستشفى نحو مجموعة العقد الناتجة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Graf merupakan teori yang ilmunya dapat diimplementasikan dalam penyelesaian masalah mencari jalur terpendek (Agung, Efendi, & Agung, 2018). Masalah jalur terpendek adalah masalah pencarian dari dua titik berupa titik awal dan titik tujuan yang diselesaikan dengan suatu algoritma, kemudian jalur yang dihasilkan dijumlahkan jaraknya sehingga akan menghasilkan jalur terpendek. Jalur terpendek berupa jalur titik awal menuju titik tujuan dengan jarak yang paling minimum. Sehingga permasalahan jalur terpendek dapat dicari solusinya dengan suatu algoritma dan menghasilkan jarak yang minimum (Siswanto, 2006).

Masalah jalur terpendek dapat dengan mudah terselesaikan dengan suatu Algoritma. Terdapat beberapa algoritma, antara lain Algoritma Dijkstra, Algoritma Bellman-Ford, Algoritma A-Star, dan Algoritma Floyd-Warshall (Usman, Wiranto, & Nurwan, 2022). Algoritma Floyd-Warshall adalah algoritma yang mudah diterapkan yang dipopulerkan dan dikembangkan oleh Warshall. Dermawan mengatakan bahwa dengan menggunakan Algoritma Floyd-Warshall akan memperoleh jalur terbaik sebagai pemecah masalah jalur terpendek (Dermawan, 2019). Dapat dikatakan algoritma terbaik karena merupakan algoritma yang dinamis. Algoritma yang dinamis adalah sebuah penyelesaian suatu masalah dimana dalam menentukan solusi dengan mengambil solusi yang telah diperoleh dari keputusan yang berkesinambungan. Masalah yang dapat diselesaikan dengan melakukan pencarian jalur terpendek salah satunya yaitu masalah kecelakaan lalu lintas. Kecelakaan lalu lintas adalah peristiwa tabrakan

yang terjadi di jalan raya menyebabkan kerusakan pada kendaraan, luka pengendara, bahkan kematian (Saputra A. D., 2017). Kecelakaan dapat dikelompokkan dalam tiga kategori yaitu kecelakaan lalu lintas berat, sedang, dan ringan. Kecelakaan ringan terjadi kerusakan pada kendaraan, sedangkan kecelakaan sedang berupa luka ringan dan kerusakan pada kendaraan. Untuk kecelakaan berat yaitu meninggalnya korban atau luka berat (Saputra A. D., 2017).

Menurut data pada halaman berita Suara Merdeka 2022, bahwa jumlah angka kecelakaan lalu lintas di Indonesia meningkat, salah satunya adalah di provinsi Jawa Tengah. Angka kecelakaannya meningkat di Kabupaten Boyolali, Jawa Tengah. Menurut Kapolres Boyolali, AKBP Morry Ernand bahwa, sebesar 55 % tahun 2021 angka kenaikan kecelakaan lalu lintas yang meninggal dunia. Jumlah kecelakaan di 2020 sebanyak 393 sedangkan tahun 2021 menjadi 790 sehingga 101 % kenaikannya. Korban yang meninggal 26 tahun 2020, sedangkan tahun 2021 73 orang sehingga meningkat 55,3 % (Murdowo, 2022). Angka tersebut menandakan bahwa perlu penindakan yang lebih serius terkait angka kematian meningkat 101% akibat kecelakaan lalu lintas di Boyolali, Jawa Tengah. Tindakan yang lebih serius untuk meminimalisir tingginya angka kecelakaan lalu lintas di Boyolali, Jawa Tengah dapat dilakukan dengan identifikasi faktor penyebabnya. Salah satu faktor tingginya angka kecelakaan tersebut karena terlambatnya penanganan dari korban kecelakaan. Keterlambatan penanganan dapat terjadi karena korban kecelakaan tidak segera dievakuasi dan dibawa ke rumah sakit dengan cepat. Maka dari itu tindakan untuk meminimalkan tingginya angka kecelakaan salah satunya dengan mencari jalur evakuasi korban kecelakaan

lalu lintas dari titik rawan kecelakaan di Boyolali, Jawa Tengah menuju rumah sakit terdekat dengan jarak terkecil.

Proses evakuasi korban kecelakaan lalu lintas di Boyolali, Jawa Tengah dilakukan oleh masyarakat yang mengetahui kecelakaan di Tempat Kejadian Perkara (TKP). Jika kecelakaan yang menimpa korban meninggal dunia, masyarakat cenderung takut untuk melakukan evakuasi sendiri, sehingga masyarakat akan menghubungi pihak rumah sakit untuk mengirimkan *ambulance*. Namun, jika korban kecelakaan belum meninggal, masyarakat akan melakukan evakuasi dengan mencari kendaraan baik mobil atau motor yang dapat membawa korban kecelakaan agar segera sampai di rumah sakit terdekat. Proses evakuasi korban kecelakaan sering terkendala karena kemacetan lalu lintas. Maka dari itu, evakuasi dapat lebih cepat jika jalur yang dipilih adalah jalan yang tidak macet lalu lintasnya.

Mencari jalur evakuasi korban kecelakaan lalu lintas dari titik rawan kecelakaan di Boyolali, Jawa Tengah menuju rumah sakit terdekat dengan mempertimbangkan jarak dan kemacetan lalu lintas. Mencari jalur evakuasi merupakan masalah jalur terpendek. Penentuannya dengan setiap lokasi titik kecelakaan direpresentasikan dalam sebuah graf, kemudian titik dapat dihubungkan satu dengan yang lainnya. Jika telah terhubung beberapa titik tersebut, maka dapat dilakukan pencarian jalur terpendek sesuai titik kecelakaan menuju rumah sakit yang terdekat dari titik lokasi kecelakaan. Sehingga, jalur evakuasi dapat diperoleh dengan mempertimbangkan jarak dan kemacetan lalu lintas jalan di Boyolali, Jawa Tengah.

Permasalahan tersebut merupakan masalah yang serius dan perlu sebuah solusi. Salah satu solusi dalam penyelesaian tersebut adalah dengan penentuan jalur terpendek. Allah SWT menerangkan bahwa permasalahan dalam kehidupan selalu ada solusinya. Sesuai dengan firman Allah dalam Al-Qur'an surat Al-Insyirah ayat 5-6 yang artinya:

“Maka, sesungguhnya beserta kesulitan ada kemudahan. Sesungguhnya beserta kesulitan ada kemudahan.” (QS. Al-Insyirah/5-6:94).

Menurut tafsir Al-Azhar (Hamka, 2015) bahwa setiap dari kita akan ditimpa suatu masalah. Tidak terkecuali Nabi Muhammad yang memiliki permasalahan yang dating berganti-ganti. Namun di samping beratnya masalah, atau beserta dengan beratnya beban, namanya diangkat Tuhan ke atas atau dimuliakan. Kesulitan selalu beserta kemudahan. Yang sulit saja tidak ada. Yang mudah saja pun tidak ada! Dalam susah berisi senang, dalam senang berisi susah, itulah perjuangan hidup. Dan ini dapat diyakinkan oleh orang-orang yang telah mengalami.

Pada ayat selanjutnya diulang kembali yaitu

“Sesungguhnya beserta kesulitan itu ada kemudahan.”

Demikian permasalahan akan terjadi terus, berulang-ulang, kesulitan itu senantiasa disertai kemudahan; dalam susah ada mudahnya, dalam sempit ada lapangnya. Bahaya yang mengancam adalah menjadi sebab akal berjalan, fikiran mencari jalan keluar. Oleh sebab itu dapatlah diyakinkan bahwa kesukaran, kesulitan, kesempitan, marabahaya yang mengancam dan berbagai ragam pengalaman hidup yang pahit, dapat menyebabkan manusia bertambah pengalaman menghadapi semuanya (Hamka, 2015).

Algoritma Floyd-Warshall digunakan di beberapa penelitian sebelumnya. Pada penelitian Arfananda membahas tentang penentuan jalur perjalanan Kota Bandung menggunakan Algoritma Floyd-Warshall, dengan menemukan solusi optimal bagi wisatawan dalam pemilihan jalur perjalanan wisata (Arfananda, Nasution, & Setianingsih, 2020). Pada penelitian tersebut memiliki masalah dengan waktu yang terbatas ketika berkunjung ke wisata Bandung kemacetan lalu lintas, jarak, dan banyaknya destinasi wisata menjadi masalah bagi wisatawan yang bepergian. Pemilihan rute yang optimal adalah solusi untuk masalah tersebut. Data kemacetan dan jarak diproses menggunakan Metode Pembobotan Aditif Sederhana (SAW). Berdasarkan hasil uji Algoritma Floyd-Warshall bahwa algoritma tersebut dapat menentukan rute terbaik yang akan dicapai beberapa tempat wisata di Kota Bandung. Dengan menggunakan Algoritma Floyd-Warshall, pengujian menggunakan perbandingan waktu komputasi di masing-masing perubahan tempat wisata sebagai titik. Perubahan waktu komputasi semakin tinggi dengan jumlah node. Berdasarkan hasil Algoritma Floyd-Warshall, uji jumlah tempat wisata yang menghitung jumlah tempat wisata dari 2 hingga 5 menghasilkan perhitungan waktu rata-rata 3 hingga 5 detik. Semakin banyak tempat wisata yang diuji maka waktu komputasi lebih lama untuk menghasilkan jalur terpendek. Rute terbaik agar dapat memberikan solusi bagi wisatawan untuk memilih rute terbaik. Oleh karena itu, bobot dan titik mempengaruhi hasil Algoritma Floyd-Warshall. Penelitian ini berhasil menampilkan rute TSP (*Travelling Sales Problem*).

Yudha sebelumnya telah meneliti jalur terpendek menggunakan Algoritma Floyd-Warshall (Yudha, Rahmani, & Purwani, 2022). Situasi pandemi membuat

masyarakat kewalahan dalam melakukan banyak hal di mana salah satunya berada di bidang pendidikan. Yang akan datang ketika sistem *full face-to-face* atau *hybrid learning* dapat diterapkan. Masalah menemukan jalan terpendek menuju masing-masing fakultas di Universitas Padjadjaran akan muncul kembali, karena banyaknya para mahasiswa yang belum pernah merasakan lingkungan di Universitas Padjadjaran. Penelitian membahas pencarian jalur terpendek dari titik awal ke masing-masing fakultas atau titik tujuan, dan jarak setiap titik dihitung menggunakan Algoritma Floyd-Warshall. Jarak terpendek ke masing-masing fakultas di Universitas Padjadjaran diperoleh dengan menggunakan Algoritma Floyd-Warshall dengan Python. Algoritma Floyd-Warshall sangat cocok untuk menemukan jalur terpendek dari titik manapun ke tujuan, karena digunakan untuk menemukan jalur terpendek antara pasangan yang mungkin di tempat yang berbeda lokasi (masalah jalur terpendek semua pasangan).

Penelitian lain yang membahas lintasan terpendek oleh Usman, tentang pendistribusian listrik di PLN Kota Gorontalo. Penelitian ini berkaitan dengan penerapan Algoritma Floyd-Warshall dalam optimalisasi rute jaringan distribusi listrik di Kota Gorontalo. Optimalisasi rute dimulai dengan mewakili tiang listrik dan panjang kabel ke dalam graf. Graf yang digunakan adalah graf berbobot dimana jalan (terkait dengan panjang kabel) direpresentasikan sebagai sisi berbobot dan tiang listrik direpresentasikan sebagai ujung. Graf ini terdiri dari satu set tiang listrik dengan total 40 titik dan satu set jalan (panjang kabel) berjumlah 46 sisi. Berdasarkan hasil perhitungan, maka diperoleh pohon rentang minimum dengan menggunakan Algoritma Floyd-Warshall plus, diperoleh bobot pohon rentang minimal (Usman, Susanti; Wiranto, Ifan; Nurwan, 2022).

Berdasarkan latar belakang tersebut, penulis akan melakukan penelitian dengan penentuan jalur terpendek dari titik rawan kecelakaan menuju rumah sakit terdekat di Boyolali, Jawa Tengah. Sehingga, dengan pencarian jalur terpendek dapat memberikan manfaat supaya jalur evakuasi korban kecelakaan dapat lebih efisien dalam waktu dan jarak, serta korban lebih cepat terselamatkan. Sehingga korban kecelakaan lebih cepat terselamatkan, diharapkan dapat menurunkan angka kematian akibat kecelakaan lalu lintas, di Boyolali Jawa Tengah.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka permasalahan dalam penelitian ini adalah bagaimana penerapan jalur terpendek dari titik rawan kecelakaan menuju rumah sakit terdekat di Boyolali, Jawa Tengah menggunakan Algoritma Floyd-Warshall.

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah mengetahui hasil penerapan jalur terpendek dari titik rawan kecelakaan menuju rumah sakit terdekat di Boyolali, Jawa Tengah menggunakan Algoritma Floyd-Warshall.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan edukasi dan informasi kepada masyarakat Boyolali dan sekitarnya serta instansi terkait mengenai jalur terdekat dari jalan rawan kecelakaan rumah sakit menuju.

1.5 Batasan Masalah

Agar masalah dapat terselesaikan dengan optimal, diperlukan batasan-batasan masalah pada permasalahan yang dibahas. Batasan masalah dari penelitian ini, yaitu:

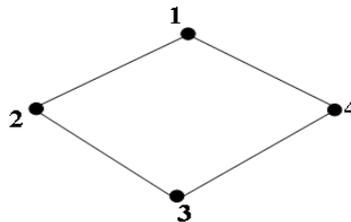
1. Rumah Sakit yang berada di sekitar jalan titik rawan kecelakaan yang diambil dari google maps.
2. Jalan yang digunakan adalah 7 jalan arteri yang rawan kecelakaan di Boyolali, Jawa Tengah yang diambil dari berita Solopos (Faizah, 2022).
3. Diasumsikan titik yang satu dengan yang lain berlaku dua arah dan jaraknya dianggap sama, seperti $A \rightarrow B$ sama jaraknya dengan $B \rightarrow A$.
4. Korban kecelakaan lalu lintas yang berada di 7 jalan arteri yang dievakuasi menuju rumah sakit terdekat.
5. Jalan yang dipilih adalah jalan dua arah, sehingga graf yang akan digunakan berbentuk graf yang memiliki arah dan berbobot.

BAB II KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Definisi Graf

Graf sesuatu himpunan tak kosong, yang dinotasikan dengan $G = (V, E)$ (Juhari, 2021). Himpunan V adalah suatu himpunan titik yang tidak boleh kosong, harus ada minimal satu titik. Contoh dari himpunan titik yaitu $V = \{v_1, v_2, \dots\}$. Himpunan E adalah suatu himpunan sisi sedemikian sehingga setiap sisi e_k dihubungkan dengan suatu pasangan tak-terurut (v_i, v_j) . Titik-titik ujung sisi e_k berkaitan dengan titik v_i, v_j , dimana i, j, k adalah bilangan $1, 2, 3, \dots, n$. Contoh dari himpunan sisi yaitu $E = \{e_1, e_2, \dots\}$. Dapat dilihat graf G pada Gambar 2.1 yaitu:



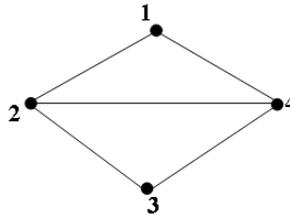
Gambar 2.1 Graf G

Gambar 2.1 di atas adalah sebuah Graf G yang terdiri dari himpunan V dan himpunan E . Himpunan titik pada Graf G yaitu $V = \{1, 2, 3, 4\}$. Himpunan sisi pada Graf G yaitu $E = \{(1, 2), (1, 4), (2, 3), (3, 4)\}$. Graf secara umum menurut Juhari dapat dikelompokkan menjadi dua jenis, yaitu:

1. Graf sederhana (simple graph)

Graf sederhana adalah graf yang tidak terdapat gelang atau sisi rangkap. Gelang adalah titik yang berawal dan berakhir pada titik yang sama. Dapat didefinisikan graf ganda $G = (V, E)$ terdiri dari himpunan tak kosong titik-

titik dan E adalah himpunan pasangan tak-terurut yang berbeda dan tidak mengandung sisi rangkap. Pada graf sederhana, sisi adalah pasangan tak-terurut, sehingga dapat ditulis (u, v) sama dengan (v, u) dimana u dan v adalah dua titik yang terhubung sehingga membentuk sisi. Seperti pada Gambar 2.2 berikut.

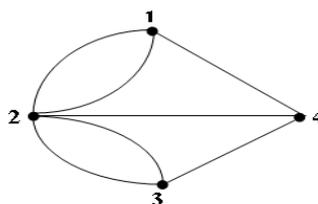


Gambar 2.2 Graf Sederhana

Gambar 2.2 diatas adalah contoh graf sederhana. Graf sederhana tersebut tidak memiliki gelang. Adapun sisi pada graf G tersebut yaitu $E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$ atau sama dengan $E = \{(2,1), (3,1), (3,2), (4,2), (4,3)\}$.

2. Graf tak-sederhana (unsimple-graph)

Graf tak-sederhana adalah graf yang terdapat gelang atau sisi rangkap. Sisi rangkap yang menghubungkan sepasang titik pada graf dapat lebih dari dua buah. Sisi ganda dapat diartikan sebagai pasangan tak-terurut yang sama. Dapat didefinisikan graf ganda $G = (V, E)$ terdiri dari himpunan tak kosong titik-titik dan E adalah himpunan ganda yang mengandung sisi rangkap. Seperti pada Gambar 2.3 yaitu:



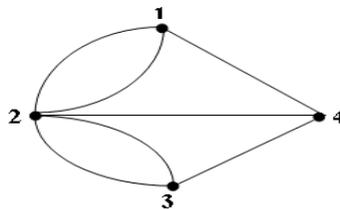
Gambar 2.3 Graf Tak-Sederhana

Gambar 2.3 diatas adalah contoh graf tak-sederhana (unsimple-graph). Graf tak-sederhana tersebut memiliki sisi rangkap yaitu $e_1 = (1,2)$ dan $e_2 = (2,1)$ dinamakan sisi rangkap, karena kedua sisi menghubungkan dua buah titik yang sama yaitu titik 1 dan 2. Kemudian, pada tersebut masih terdapat sisi rangkap yaitu sisi $e_3 = (2,3)$ dan $e_4 = (3,2)$ yang sama-sama dihubungkan oleh titik 2 dan 3.

Orientasi arah adalah ada tidaknya arah pada sisi graf. Berdasarkan orientasi arah dikelompokkan menjadi dua jenis, yaitu:

1. Graf tak-berarah (undirected graph)

Graf tak-berarah adalah sisinya tidak mempunyai arah pada sisi graf. Pada graf ini urutan pasangan titik yang dihubungkan oleh sisi tidak diperhatikan. Sehingga $(u, v) = (v, u)$ adalah sisi yang sama.

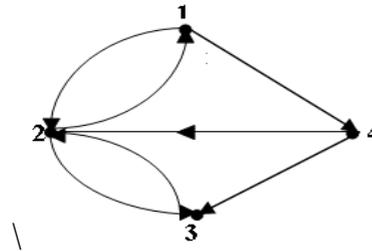


Gambar 2.4 Graf Tak-Berarah

2. Graf berarah (directed graph)

Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. pada graf berarah (u, v) dan (v, u) menyatakan dua busur yang berbeda, dengan kata lain (u, v) tidak sama dengan (v, u) . Untuk sisi (u, v) titik u dinamakan titik asal u dan titik v dinamakan titik terminal. Graf berarah sering dipakai untuk menggambarkan aliran proses, peta lalu lintas suatu kota (jalan searah atau dua arah), dan sebagainya. Pada graf berarah gelang

diperbolehkan, tetapi sisi ganda tidak.



Gambar 2.5 Graf Berarah

Definisi graf dapat diperluas sehingga mencakup graf ganda berarah (directed multigraph). Pada gerak ganda berarah gelang dan sisi ganda diperbolehkan ada tabel di bawah akan meringkas perluasan definisi graf.

Tabel 2.1 Perbedaan Graf Secara Ringkas

| Jenis | Sisi | Sisi ganda diperbolehkan? | Sisi Gelang diperbolehkan |
|--------------------|---------------|---------------------------|---------------------------|
| Graf Sederhana | Tidak-berarah | Tidak | Tidak |
| Graf Ganda | Tidak-berarah | Ya | Tidak |
| Graf Berarah | Berarah | Tidak | Ya |
| Graf-ganda berarah | Berarah | Ya | Ya |

Pada Tabel 2.1 diatas, graf yang pertama adalah graf sederhana, yang kedua adalah yang setiap elemennya termasuk sisi ganda atau gelang disebut graf semu. Sedangkan adalah graf dengan komponen tunggal yang berasal dari komponen pendamping. Graf ketiga-tiganya ini tidak memiliki orientasi arah.

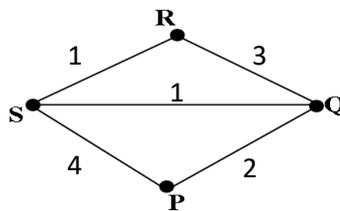
Istilah yang sering digunakan dan berkaitan dengan graf yaitu

1. Bertetangga

Bertetangga adalah apabila sepasang titik itu yang dihubungkan oleh sisi. Matriks ketetanggaan digunakan pada graf sederhana yang tersusun dari bilangan nol dan satu yang berukuran persegi. Baris dan kolom adalah implementasi pada graf berasal dari titik atau titik (Adiwijaya, 2016). Contoh nya yaitu jika dimisalkan a_{ij} adalah unsur pada matrik, sehingga:

- Jika $a_{ij} = 1$ sehingga titik i dan titik j bertetangga.
- Jika $a_{ij} = 0$ sehingga titik i dan titik j tidak bertetangga.

Seperti pada Gambar 2.6 berikut:



Gambar 2.6 Contoh Matriks Ketetanggaan

Berikut adalah Tabel 2.2 bertetangga tidaknya titik satu dengan yang lain sesuai pada Gambar 2.6 berikut ini:

Tabel 2.2 Tabel Contoh Graf Ketetanggaan

| | P | Q | R | S |
|---|---|---|---|---|
| P | 0 | 2 | 3 | 4 |
| Q | 2 | 0 | 3 | 1 |
| R | 3 | 3 | 0 | 1 |
| S | 4 | 1 | 1 | 0 |

2. Jalur (path)

Jalur (path) yang panjangnya n dari titik awal v_0 menuju titik tujuan v_n pada graf G adalah barisan yang berselang-seling titik-titik dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ yaitu sisi-sisi pada graf G .

3. Graf berbobot (weighted graph)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah bobot yaitu jarak antara dua titik yang membentuk sisi.

2.1.2 Matriks

Bentuk persegi panjang terdiri dari bilangan-bilangan bulat berturut-turut disebut sebagai matriks. Rentang angka dikenal sebagai entri matriks (Anton H., 2000). Jadi, matriks adalah pengelompokan bilangan yang ukurannya persegi atau persegi panjang, tersusun dari baris dan kolom, dan berisi dari susunan bilangan. Yang bisa dituslikan dengan menggunakan tanda kurung atau tanda kurung siku digunakan untuk melampirkan angka, yang disebut sebagai elemen matriks. Jumlah baris dan kolom, juga dikenal sebagai urutan matriks, menentukan ukuran matriks. Bentuk matriks adalah seperti berikut ini, yaitu:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Dapat ditulis dengan matriks dengan ukuran $A_{m \times n}$. Adapun keterangannya yaitu

A = Nama matriks

m = Banyak baris pada matriks

n = Banyak kolom pada matriks

$m \times n$ = Ordo pada matriks

Beberapa matriks yang digunakan pada Graf antara lain yaitu:

2.1.3 Persegi

Matriks sering disebut matriks persegi adalah matriks dengan ukuran yang mempunyai baris dan kolom dengan jumlah yang sama.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

2.1.4 Simetri

Matriks Simetri adalah tersusun dari setiap elemen yang simetri antara diagonal utama dengan nilai yang memiliki nilai yang sama dengan transposenya ($A = A^T$), dimana matriks ini berbentuk matriks persegi.

$$A = \begin{bmatrix} 0 & 2 & 5 \\ 2 & 0 & 3 \\ 5 & 3 & 0 \end{bmatrix}$$

2.1.3 Algoritma

Algoritma secara umumnya merupakan cara dan urutan langkah atau tahapan yang diatur secara sistematis untuk menyelesaikan suatu permasalahan. Sehingga proses yang akan diselesaikan dapat sesuai dengan algoritma atau urutan langkah yang telah disusun. Pada prinsipnya algoritma dari sebuah proses ada tiga tahapan seperti Gambar 2.7.



Gambar 2.7 Prinsip Algoritma

Terdapat inputan yang dimasukkan ke dalam proses dan proses akan menghasilkan output atau keluaran. Beberapa kriteria yang harus dimiliki oleh suatu algoritma sehingga bisa dikatakan sebagai algoritma yang baik menurut (Komarullah, 2013) adalah

1. *Input*

Suatu algoritma harus memiliki inputan baik 0 *inputan* atau lebih. jadi suatu algoritma dapat memiliki lebih dari satu inputan atau bahkan tidak

memiliki inputan sama sekali dari usernya dan data input dan didapatkan dari dalam algoritma itu sendiri.

2. *Output*

Output adalah suatu *outputan* harus dihasilkan oleh sebuah algoritma. Karena Akan menjadi sia-sia Apabila sebuah algoritma tidak menghasilkan keluaran sama sekali. Karena algoritma disusun dengan tujuan untuk menghasilkan suatu hasil atau keluaran.

3. *Finiteness*

Finiteness adalah suatu proses pasti akan berhenti ketika hasil yang diharapkan sudah tercapai. Demikian pula dengan algoritma yang dibuat harus memiliki titik akhir ketika proses yang diharapkan telah selesai dikerjakan.

4. *Definiteness*

Definiteness adalah algoritma yang dibuat tidak menimbulkan makna ganda sehingga Setiap proses yang dijalankan tidak menimbulkan multitafsir kepada pelakunya agar hasil yang dicapai sesuai dengan yang diharapkan.

5. *Effectiveness*

Effectiveness Adalah Setiap proses dalam algoritma harus dibuat seefektif mungkin. karena setiap proses dalam algoritma memerlukan waktu untuk dieksekusi sehingga kita harus memastikan Tidak ada proses yang sia-sia dalam algoritma kita seperti contohnya kita tidak perlu menambahkan garam ke dalam air laut untuk mendapatkan rasa asin dari air laut proses penambah garam ke dalam air laut adalah proses yang sia-sia.

Beberapa struktur algoritma secara umum memiliki tiga struktur dasar yaitu perulangan percabangan dan terurut sebagai berikut

1. Perulangan atau *looping*

Perulangan dalam kehidupan sehari-hari banyak pekerjaan yang dilakukan dengan hal tersebut. Contohnya yaitu kegiatan berulang seperti makan, minum, mencuci, dan lain sebagainya dimana pekerjaan tersebut akan kita lakukan sampai pada batas tertentu sehingga proses tersebut dihentikan.

2. Percabangan

Percabangan merupakan proses pemilihan proses selanjutnya pada keadaan tertentu. hal ini terjadi karena pada suatu algoritma bisa terjadi keadaan yang tidak pada semestinya sehingga kita diminta untuk menentukan proses apa yang akan dilakukan Selanjutnya apabila keadaan tersebut terjadi. misalnya kita memiliki algoritma berangkat kuliah kemudian terjadi kondisi cuaca hujan maka kita akan diminta untuk memilih proses selanjutnya “pesan taksi” atau “menggunakan mantel” atau “tidak jadi berangkat”. Pada saat cuaca “hujan” di sinilah terjadi *Branching* atau percabangan dimana kita diminta untuk memilih proses selanjutnya.

3. Terurut (*sequential*)

Terurut adalah proses urutan algoritma sesuai dengan urutannya. Algoritma ini memiliki kondisi yang ideal atau tidak terjadi kondisi perulangan atau percabangan. Sehingga, proses akan berjalan urut mulai dari proses 1 sampai dengan proses ke- n .

2.1.4 Algoritma Floyd-Warshall

Algoritma Floyd-Warshall adalah strategi pemecahan masalah yang melihat solusi sebagai serangkaian keputusan yang berhubungan. Menandakan bahwa solusi tersebut terdiri dari solusi dari langkah yang telah dilakukan, dan ada beberapa opsi. Pendekatan ini menggunakan program yang lebih dinamis dimana keberhasilan menentukan jalur terpendek antara semua pasangan titik ditentukan dengan menemukan solusi terbaik (Komarullah, 2013). Algoritma Floyd-Warshall adalah bentuk pemrograman, dimana penyelesaian masalah dengan mengambil solusi yang mana dijadikan suatu keputusan (Cormen, 2003).

Jalur terpendek diantara semua pasang titik dapat dicari dengan graf berbobot dan berarah dengan menggunakan Algoritma Floyd-Warshall. Cara kerja algoritma tersebut yaitu perbandingan dari semua kemungkinan jalur pada graf di setiap pasang titik dan kombinasi titik yang dihasilkan akan ditinjau kembali. Bentuk Algoritma Floyd-Warshall yang menggunakan graf berbobot dan berarah dapat ditulis sebagai berikut:

1. $M = M_0$

2. Untuk $k = 1$ hingga n , jalankan:

Untuk $i = 1$ hingga n , jalankan:

Untuk $j = 1$ hingga n , jalankan:

Jika $M_{[i,j]} > M_{[i,k]} + M_{[k,j]}$ sehingga $M_{[i,j]}$ lebih besar maka dapat diganti dengan $M_{[i,k]} + M_{[k,j]}$ yang lebih kecil bobotnya.

3. $M^* = M$

Keterangannya adalah:

$$M = \text{Matriks}$$

M_0 = Matriks hubung graf mula-mula

k = Iterasi 1 sampai ke- n

n = Jumlah elemen dalam matriks

i = Titik awal pada v_i

j = Titik akhir pada v_j

M^* = Hasil matriks setelah perbandingan

Algoritma Floyd-Warshall menghasilkan n matriks untuk setiap iterasi dalam proses pencarian rute terpendek. Meskipun deteksi hitungan manual membutuhkan proses yang lambat karena setiap elemen harus diidentifikasi secara terpisah, algoritma Floyd-Warshall memiliki keunggulan deteksi hitungan yang lebih cepat berkat penggunaan perbandingan. Selain itu, algoritma Floyd Warshall dapat diimplementasikan dengan cukup mudah.

Menurut (Aini & Salehipour, 2011) terdapat langkah-langkah pada Algoritma Floyd-Warshall, yaitu:

1. Tetapkan nilai titik awal ke nilai titik tujuan.
2. Tuliskan jarak sisi antara titik asal dan titik berikutnya.
3. Pada titik target, tambahkan jarak tepi untuk setiap titik yang berurutan.
4. Untuk setiap jalur layak yang ditemukan, bandingkan nilai tepinya.
5. Pilih jalur dengan nilai sisi terkecil (optimal).

Misalkan terdapat suatu graf G dengan banyak titik dinotasikan V yang pada setiap titik benomor 1 sampai dengan n . Misalkan jika ada suatu fungsi *shortest path* (i, j, k) yang mengembalikan kemungkinan jalur terpendek dari i ke j dengan hanya memanfaatkan titik 1 s.d. k sebagai titik perantara.

Tujuan dari penggunaan ketiga fungsi ini adalah untuk menemukan jalur yang paling pendek antara masing-masing titik i dan titik j menggunakan titik 1 sampai dengan $k + 1$ perantara. Ada dua kemungkinan yang akan terjadi, yaitu:

1. Jalur terpendek yang sebenarnya hanya berasal dari titik-titik yang berada antara 1 hingga k .
2. Ada sebagian jalur yang berasal dari titik-titik 1 hingga $k + 1$, dan juga $k + 1$ hingga j .

Perlu diketahui bahwa jalur terpendek dari i ke j yang hanya melewati titik 1 s.d. k telah didefinisikan pada fungsi *shortest path* (i, j, k) dan telah jelas bahwa jika ada solusi dari i s.d. $k + 1$ hingga j , maka panjang dari solusi tadi adalah jumlah dari jalur terpendek dari i s.d. $k + 1$ (yang melewati titik-titik 1 s.d. k), dan jalur terpendek dari $k + 1$ s.d. j juga menggunakan titik-titik dari 1 s.d. k .

Algoritma Floyd-Warshall yang digunakan untuk membantu memecahkan masalah lebih memandang solusi akhir yang akan diperoleh sebagai suatu keputusan yang saling terkait. Sehingga, solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya. Adapun kelebihan dari algoritma flood warsoul antara lain:

1. Algoritma Floyd Warshall dapat digunakan untuk mencari jarak terpendek atau shortcut dari setiap pasang titik.
2. Algoritma Floyd Warshall menggunakan matriks bobot $n \times n$ sebagai masukan dimana n merupakan jumlah titik.
3. Algoritma Floyd Warshall dapat mentolerir *negative edge*.

4. Looping yang dihasilkan membantu menemukan bobot jarak yang belum diketahui sehingga dapat dicari jalur terpendek.

2.2 Kajian Integrasi Dengan Al-Qur'an Dan Hadits

Islam sebagai agama rahmatan lil 'Alamin telah mengatur dan menjelaskan segala aspek tentang tata cara yang baik dalam kehidupan manusia, salah satunya tolong-menolong. Islam sangat berperan penting sebagai agama yang mulia serta Al-Qur'an dan hadits yang berperan sebagai pedoman hidup dan pendukung dalam menjalani kehidupan yang sesuai dengan syari'at islam. Berpegang teguh kepada Al-Qur'an dan hadits adalah suatu kewajiban dan hal tersebut akan sangat berpengaruh dalam kehidupan kita, baik dalam bersikap maupun menyelesaikan masalah yang ada. Pada zaman yang serba modern yang serba ada dan mudah ini, permasalahan kehidupan semakin bertambah. Salah satunya adalah permasalahan tingginya angka kecelakaan di Boyolali, Jawa Tengah. Sebagai manusia kita hendaknya membantu permasalahan tersebut sesuai dengan kemampuan masing-masing. Seperti yang tercantum dalam Al-Qur'an surat Al-Maidah ayat 2 tentang perintah saling tolong-menolong yang artinya:

"... Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksa-Nya." (QS. Al-Maidah /06:02).

Menurut tafsir Al-Azhar (Hamka, 2015) bahwa kita diperintahkan supaya hidup tolong-menolong, dalam membina *Al-Birru*, yaitu segala sesuatu yang mengandung kebaikan dan bermanfaat. Tolong menolong yang dimaksud adalah tolong menolong yang tidak menimbulkan suatu perbuatan dosa dan permusuhan

dan menyakiti sesama manusia. Berpegang berdasarkan kepada menegakkan Takwa, yaitu mempererat hubungan dengan Tuhan.

Kecelakaan Boyolali dapat ditekan angka kecelakaannya. Dengan berusaha salah satunya dengan mencari jalur evakuasi terdekat dari titik rawan kecelakaan menuju rumah sakit yang terdekat. Seberapa berat dan kompleks suatu masalah sebagai manusia kita harus tetap berusaha atau *tawakal*. Seperti yang dijelaskan pada firman Allah SWT dalam Al-Qur'an Surah Yusuf ayat 87 yang artinya.

“Hai anak-anakku, pergilah kamu, maka carilah berita tentang Yusuf dan saudaranya dan jangan kamu berputus asa dari rahmat Allah. Sesungguhnya tiada berputus asa dari rahmat Allah, melainkan kaum yang kafir.”(Q.S Yusuf:87)

Ayat di atas juga berisi tentang larangan untuk tidak berputus asa dari rahmat Allah SWT. Yakin dan percaya kepada Allah memang dapat membangkitkan harapan. Maka dari itu, kesedihan yang dialami Ya'qub tidak mampu membuatnya putus asa bahwa kedua anaknya yang hilang pasti akan kembali ke pangkuannya. Nalurnya mengatakan bahwa kedua anaknya itu masih hidup, dan bahwa pertemuannya dengan mereka sudah semakin dekat. Ya'qub kemudian meminta anak-anaknya pergi ke Mesir mencari kedua anaknya yang hilang itu dengan mengatakan, “Anak-anakku, pergilah kalian ke Mesir dan bergabunglah dengan kakakmu lalu carilah Yusuf dan saudaranya, Benyamin. Tanyakan kepada orang-orang tentang mereka secara lemah lembut tanpa harus dirasakan orang. Jangan berputus asa terhadap sifat kasih sayang Allah yang pasti akan mengembalikan mereka kepada kita. Sebab, sesungguhnya yang berputus asa terhadap kasih sayang Allah hanyalah orang-orang yang ingkar dan kafir.”

2.3 Kajian Topik Dengan Teori Pendukung

Rute terpendek dianalisis pada jalan titik rawan kecelakaan di Boyolali, Jawa Tengah menuju Rumah sakit terdekat dengan lokasi kecelakaan dengan menggunakan Algoritma Floyd-Warshall. Rute tersebut dianalisis agar memperoleh rute yang dapat digunakan untuk mengevakuasi korban kecelakaan lebih cepat sampai menuju Rumah sakit. Sehingga dapat diharapkan angka kematian dapat diminimalisir atau dikurangi karena penanganan yang lebih cepat.

Algoritma Floyd-Warshall terdiri dari sejumlah langkah, mirip dengan mempelajari sistem koordinat untuk lokasi yang akan digunakan sebagai objek, memodelkan data yang diperoleh ke dalam graf penelitian, menafsirkan graf yang dihasilkan sebagai matriks M yang mengukur $n \times n$ dimana n adalah jumlah titik yang elemennya mengandung bobot samping atau jarak antar titik, menghitung kecil elemen dibanding dengan elemen $M_{[i,j]}$ dengan $M_{[i,k]} + M_{[k,j]}$ sampai iterasi $k = n$, pengujian rute terpendek dilakukan dengan memasukkan titik awal atau i dan titik tujuan atau j dengan disesuaikan apa yang menjadi kebutuhan.

Penelitian ini akan membahas lebih dalam berkaitan dengan algoritma Floyd-Warshall dengan menggambarkan rute terpendek dari titik rawan kecelakaan di Boyolali, Jawa Tengah menuju rumah sakit terdekat. Ada banyak metode dan algoritma untuk mencari rute terpendek pada graf; yang paling optimal dari ini adalah algoritma Floyd-Warshall. Kasus untuk permasalahan tersebut masih jarang dibahas menggunakan metode Floyd-Warshall. Oleh karena itu, peneliti mempertimbangkan algoritma Floyd-Warshall untuk menemukan rute terpendek dari 13 lokasi titik rawan kecelakaan di Boyolali, Jawa Tengah dan 10 rumah sakit yang digunakan. Algoritma Floyd-Warshall adalah metode

pemrograman untuk memecahkan masalah dengan menerapkan solusi yang ditentukan sebagai faktor penyebab yang paling jelas. Ini menunjukkan bahwa solusi saat ini dibangun berdasarkan solusi sebelumnya dan memungkinkan lebih dari satu solusi. Sehingga solusi tersebut berupa semua jalur yang mungkin sehingga jalur yang diperoleh lebih dari satu, yang nantinya akan pilih jalur dengan jarak terkecil.

BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif yang didominasi dengan suatu perhitungan (Shahid, 2021). Penelitian menggunakan pendekatan studi lapangan melalui *Google Maps*. Studi penelitian evaluasi kuantitatif adalah studi yang meneliti sistem yang berfungsi dengan tujuan menentukan seberapa baik kinerjanya ketika digunakan. Melakukan studi literatur dengan menemukan sumber informasi dan mengumpulkan data dari item yang sedang diteliti. Melalui tahap pemrosesan data, setelah itu pengujian kuantitatif kemudian didapatkan hasil data penelitian secara kuantitatif diskriptif (berdasarkan perhitungan dan penjelasan mendetail).

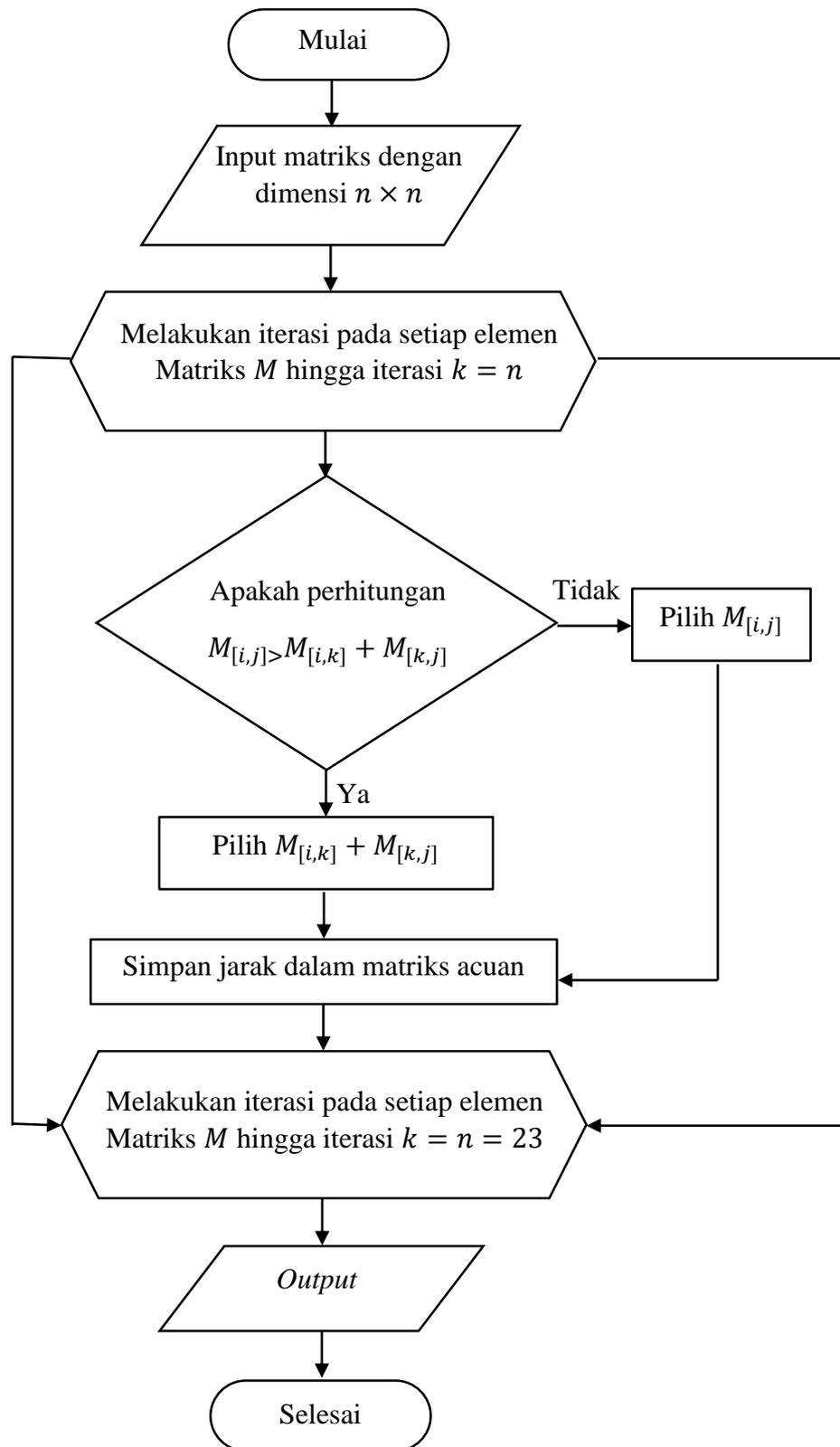
3.2 Tahapan Penelitian

Penyelesaian masalah memiliki beberapa tahapan. Tahapan penelitian ini sebagai berikut:

1. Menghimpun data dari *google maps* berupa capturing *Google Maps*.
2. Memodelkan data jarak yang telah diperoleh ke dalam bentuk graf penelitian, titik lokasi jalan rawan kecelakaan dan persimpangan akan diberikan pelabelan menggunakan huruf dan angka sedangkan jarak antar titik ditandai dengan sisi dan diberi bobot jarak antar titik tersebut.
3. Menginterpretasikan graf yang dihasilkan sebagai matriks M berukuran $n \times n$, dimana n adalah jumlah simpul yang elemennya berisi bobot sisi atau jarak antar simpul.

4. Melakukan perhitungan dengan cara menghitung setiap elemen terkecil dengan cara membandingkan elemen $M_{[i,j]}$ dengan $M_{[i,k]} + M_{[k,j]}$ sampai iterasi $k = n$ untuk mengisi setiap elemen dari matriks M yang belum diketahui jaraknya. Iterasi tersebut akan dilakukan dengan bantuan Matlab.
5. Melakukan pengujian rute terpendek dengan menginputkan titik awal (i) dan titik tujuan sesuai yang diinginkan.

3.3 Flowchart



Gambar 3.1 Flowchart Algoritma Floyd-Warshall

BAB IV PEMBAHASAN

4.1 Deskripsi Data

Melakukan pengamatan dan mengumpulkan data obyek yang akan dijadikan sebagai bahan penelitian. Data yang dihimpun berupa ruas jalan rawan kecelakaan yang diambil dari Berita Solopos (Faizah, 2022). Terdapat 7 ruas jalan arteri yang rawan kecelakaan dan telah dikonfirmasi oleh Kapolres Boyolali, Jawa Tengah sesuai dokumentasi pada gambar berikut.



Gambar 4.1 Dokumentasi Konfirmasi Data di Polres Boyolali

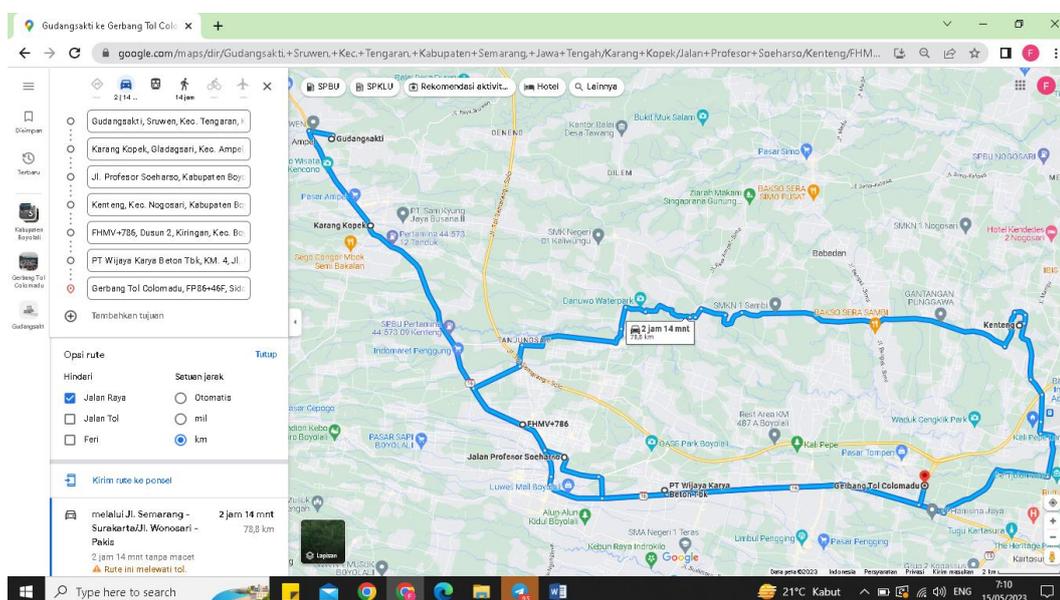
Data ruas jalan rawan kecelakaan yang dihimpun menjadi obyek penelitian yaitu:

Tabel 4.1 Obyek Penelitian Lokasi Ruas Jalan Rawan Kecelakaan

| No. | Ruas Jalan Rawan Kecelakaan | Alamat |
|-----|--------------------------------|---|
| 1. | Pertigaan Sruwen | Gudang Sakti, Urutsewu, Kec. Ampel, Kabupaten Boyolali, Jawa Tengah 50775 |
| 2. | Pertigaan Kebon Jeruk Ampel | Karang Kopek, Gladagsari, Kec. Ampel, Kabupaten Boyolali, Jawa Tengah |
| 3. | Pertigaan Prof. Soeharso Ampel | Jl. Profesor Soeharso, Kabupaten Boyolali, Jawa Tengah |

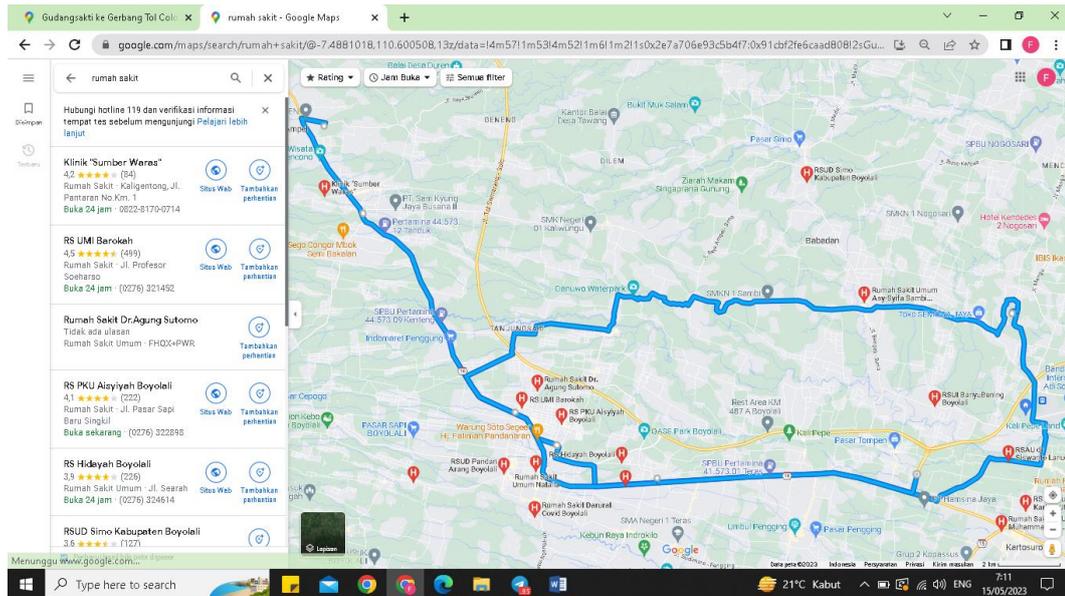
| | | |
|----|-----------------------------------|---|
| 4. | Pertigaan Kenteng, Boyolali | Kenteng, Kec. Nogosari, Kabupaten Boyolali, Jawa Tengah |
| 5. | Perempatan Terminal Lama Boyolali | FHMV+786, Dusun 2, Kiringan, Kec. Boyolali, Kabupaten Boyolali, Jawa Tengah 57314 |
| 6. | Pertigaan Wika, Mojosongo | PT Wijaya Karya Beton Tbk, KM. 4, Jl. Raya Boyolali-Semarang No.5, Pomah, Mojosongo, Kec. Mojosongo, Kabupaten Boyolali |
| 7. | Pintu Keluar Tol Ngasem | Gerbang Tol Colomadu, FP86+46F, Sidorejo, Ngasem, Kec. Colomadu |

Dibawah ini merupakan titik lokasi 7 ruas jalan arteri yang rawan kecelakaan di Boyolali menggunakan *Google Maps*:



Gambar 4.2 Titik Lokasi Kecelakaan

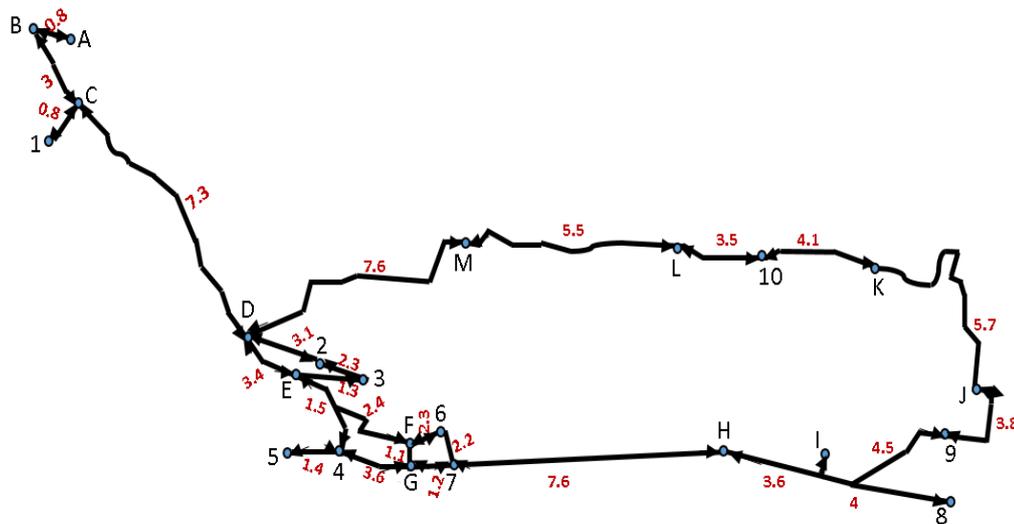
Rumah sakit yang akan dipilih adalah yang tersebar di sekitar lokasi rawan kecelakaan. Seperti yang terlihat pada gambar berikut ini:



Gambar 4.3 Lokasi Sebaran Rumah Sakit

4.2 Pengolahan Data

Berdasarkan data titik lokasi, penulis membuat graf yang dapat mewakili tiap-tiap titik dan jarak di lokasi penelitian. Pada graf tersebut titik mewakili lokasi dan sisi mewakili jarak, Algoritma Floyd-Warshall menghitung semua bobot minimum yang mungkin untuk setiap pasang titik. Rute yang dilalui digambarkan pada graf berikut



Gambar 4.4 Graf Wilayah Penelitian

Daftar titik pada graf penelitian:

Tabel 4.2 Daftar Titik dari Graf Wilayah Penelitian

| Titik | Nama Tempat |
|-------|--|
| A | Persimpangan Jl. Sruwen-Karanggede |
| B | Persimpangan Jl. Raya Salatiga-Solo |
| C | Tugu Lilin |
| D | Terminal Bus Boyolali |
| E | Warung Soto Hj. Fatimah Pandanarang |
| F | Alun-alun Lor Boyolali |
| G | Dya Hotel Boyolali |
| H | Waroeng Spesial Sambal "SS" Banyudono, |
| I | Gerbang Tol Colomadu |
| J | Kali Pepe Land |
| K | Toko Semoga Jaya |
| L | SMK N 1 Sambu |
| M | Danuwo Waterpark |
| 1 | Klinik Sumber Waras |
| 2 | RSU Umi Barokah |
| 3 | RS PKU Aisyiyah |
| 4 | RSU Natalia |
| 5 | RSUD Pandanarang |
| 6 | RS Hidayah Boyolali |
| 7 | RS Indriyanti Boyolali |
| 8 | RS PKU Muhammadiyah Kartasura |
| 9 | RS TNI Auri |
| 10 | Rumah Sakit Umum Asy-Syifa Sambu |

Dari graf tersebut, selanjutnya jarak antar titik dapat dinyatakan dalam bentuk tabel yang tertera pada **lampiran 2**. Baris pada tabel menyatakan (i) yang berisi titik awal. Kolom pada tabel tersebut menyatakan (j) yang berarti titik tujuan. Elemen pada tabel tersebut berisi jarak sisi dari (i) titik awal menuju titik tujuan (j) dalam satuan kilometer. Sementara tanda (∞) pada tabel menyatakan bahwa jarak antara (i) titik awal menuju titik tujuan (j) pada iterasi tersebut belum ditemukan.

4.3 Perhitungan Lintasan Terpendek

Berikut proses perhitungan Algoritma Floyd-Warshall menggunakan matriks M untuk menentukan jarak terpendek titik rawan kecelakaan menuju rumah sakit terdekat.

1. Iterasi $k = 0$

Matriks iterasi $k = 0$ yaitu matriks yang menyatakan nilai yang sebenarnya dari graf. Representasi matriks $M^{(0)}$ iterasi $k = 0$ dapat dilihat pada **lampiran 3**.

Dari Matriks tersebut diketahui bahwa jarak dari titik A ke titik B pada matriks adalah 0.8 km, dari titik B ke titik C adalah 3 km, dan seterusnya. Kemudian tanda " ∞ " pada matriks mempunyai arti tidak ada jalur langsung dari titik awal ke titik yang dituju. Untuk menghitung jarak terpendek maka dilakukan perhitungan sampai matriks iterasi $k = n$ pada setiap elemennya, yang dalam penelitian ini akan dilakukan perhitungan sampai matriks iterasi $k = 23$.

2. Iterasi $k = 1$

Representasi matriks $M^{(1)}$ iterasi $k = 1$ dapat dilihat pada **lampiran 4**.

Selanjutnya, untuk setiap elemen matriks M dicek apakah $M_{[i,j]} > M_{[i,k]} + M_{[k,j]}$. Jika ya, maka $M_{[i,j]}$ diganti dengan $M_{[i,k]} + M_{[k,j]}$

Beberapa hasil pengecekan pada iterasi $k = 1$ adalah sebagai berikut :

a. $M_{[3,2]} = 3$ sedangkan $M_{[3,1]} + M_{[1,2]} = \infty + 0.8 = \infty$

karena nilai $M_{[i,j]} < M_{[3,1]} + M_{[1,2]}$ maka nilai $M_{[3,2]}$ tidak berubah atau tetap bernilai 3.

b. $M_{[3,4]} = 7.3$ sedangkan $M_{[3,1]} + M_{[1,4]} = \infty + \infty = \infty$

- karena nilai $M_{[3,4]} < M_{[3,1]} + M_{[1,4]}$ maka nilai $M_{[3,4]}$ tidak berubah atau tetap bernilai 7.3.
- c. $M_{[3,14]} = 0.8$ sedangkan $M_{[3,1]} + M_{[1,14]} = \infty + \infty = \infty$
 karena nilai $M_{[3,14]} < M_{[3,1]} + M_{[1,14]}$ maka nilai $M_{[3,14]}$ tidak berubah atau tetap bernilai 0.8.
- d. $M_{[5,4]} = 3.4$ sedangkan $M_{[5,1]} + M_{[1,4]} = \infty + \infty = \infty$
 karena nilai $M_{[5,4]} < M_{[5,1]} + M_{[1,4]}$ maka nilai $M_{[5,4]}$ tidak berubah atau tetap bernilai 0.8.
- e. $M_{[13,4]} = 7.6$ sedangkan $M_{[13,1]} + M_{[1,4]} = \infty + \infty = \infty$
 karena nilai $M_{[13,4]} < M_{[13,1]} + M_{[1,4]}$ maka nilai $M_{[13,4]}$ tidak berubah atau tetap bernilai 7.6.
- f. $M_{[15,4]} = 3.1$ sedangkan $M_{[15,1]} + M_{[1,4]} = \infty + \infty = \infty$
 karena nilai $M_{[15,4]} < M_{[15,1]} + M_{[1,4]}$ maka nilai $M_{[15,4]}$ tidak berubah atau tetap bernilai 3.1.
- g. $M_{[6,5]} = 2.4$ sedangkan $M_{[6,1]} + M_{[1,5]} = \infty + \infty = \infty$
 karena nilai $M_{[6,5]} < M_{[6,1]} + M_{[1,5]}$ maka nilai $M_{[6,5]}$ tidak berubah atau tetap bernilai 2.4.
- h. $M_{[8,5]} = \infty$ sedangkan $M_{[8,1]} + M_{[1,5]} = \infty + \infty = \infty$
 karena nilai $M_{[8,5]} = M_{[8,1]} + M_{[1,5]}$ maka nilai $M_{[8,5]}$ tidak berubah atau tetap bernilai ∞ .
- i. $M_{[16,5]} = 1.3$ sedangkan $M_{[16,1]} + M_{[1,5]} = \infty + \infty = \infty$
 karena nilai $M_{[16,5]} < M_{[16,1]} + M_{[1,5]}$ maka nilai $M_{[16,5]}$ tidak berubah atau tetap bernilai 1.3.

- j. $M_{[17,5]} = 1.5$ sedangkan $M_{[17,1]} + M_{[1,5]} = \infty + \infty = \infty$
 karena nilai $M_{[17,5]} < M_{[17,1]} + M_{[1,5]}$ maka nilai $M_{[17,5]}$ tidak
 berubah atau tetap bernilai 1.5.
- k. $M_{[7,6]} = 1.1$ sedangkan $M_{[7,1]} + M_{[1,6]} = \infty + \infty = \infty$
 karena nilai $M_{[7,6]} < M_{[7,1]} + M_{[1,6]}$ maka nilai $M_{[7,6]}$ tidak
 berubah atau tetap bernilai 1.1.
- l. $M_{[9,8]} = 3.6$ sedangkan $M_{[9,1]} + M_{[1,8]} = \infty + \infty = \infty$
 karena nilai $M_{[9,8]} < M_{[9,1]} + M_{[1,8]}$ maka nilai $M_{[9,8]}$ tidak
 berubah atau tetap bernilai 3.6.
- m. $M_{[20,8]} = 7.6$ sedangkan $M_{[20,1]} + M_{[1,8]} = \infty + \infty = \infty$
 karena nilai $M_{[20,8]} < M_{[20,1]} + M_{[1,8]}$ maka nilai $M_{[20,8]}$ tidak
 berubah atau tetap bernilai 7.6.
- n. $M_{[11,10]} = 5.7$ sedangkan $M_{[11,1]} + M_{[1,10]} = \infty + \infty = \infty$
 karena nilai $M_{[11,10]} < M_{[11,1]} + M_{[1,10]}$ maka nilai $M_{[11,10]}$ tidak
 berubah atau tetap bernilai 5.7.
- o. $M_{[13,12]} = 5.5$ sedangkan $M_{[13,1]} + M_{[1,12]} = \infty + \infty = \infty$
 karena nilai $M_{[13,12]} < M_{[13,1]} + M_{[1,12]}$ maka nilai $M_{[13,12]}$ tidak
 berubah atau tetap bernilai 5.5.

3. Iterasi $k = 2$

Representasi matriks $M^{(2)}$ iterasi $k = 2$ dapat dilihat pada **lampiran 5**.

Selanjutnya, untuk setiap elemen matriks M dicek apakah $M_{[i,j]} >$

$M_{[i,k]} + M_{[k,j]}$. Jika ya, maka $M_{[i,j]}$ diganti dengan $M_{[i,k]} + M_{[k,j]}$

Beberapa hasil pengecekan pada iterasi $k = 2$ adalah sebagai berikut :

- a. $M_{[3,1]} = \infty$ sedangkan $M_{[3,2]} + M_{[2,1]} = 3 + 0.8 = 3.8$
 karena nilai $M_{[13,12]} > M_{[13,1]} + M_{[1,12]}$ maka nilai $M_{[13,12]}$
 berubah atau diganti menjadi 3.8.
- b. $M_{[4,13]} = 7.6$ sedangkan $M_{[4,2]} + M_{[2,13]} = \infty + \infty = \infty$
 karena nilai $M_{[4,13]} < M_{[4,2]} + M_{[2,13]}$ maka nilai $M_{[4,13]}$ tidak
 berubah atau tetap bernilai 7.6.
- c. $M_{[4,3]} = 7.3$ sedangkan $M_{[4,2]} + M_{[2,3]} = \infty + 3 = \infty$
 karena nilai $M_{[4,3]} < M_{[4,2]} + M_{[2,3]}$ maka nilai $M_{[4,3]}$ tidak
 berubah atau tetap bernilai 7.3.
- d. $M_{[14,3]} = 0.8$ sedangkan $M_{[14,2]} + M_{[2,3]} = \infty + 3 = \infty$
 karena nilai $M_{[14,3]} < M_{[14,2]} + M_{[2,3]}$ maka nilai $M_{[14,3]}$ tidak
 berubah atau tetap bernilai 0.8.
- e. $M_{[1,3]} = \infty$ sedangkan $M_{[1,2]} + M_{[2,3]} = 0.8 + 3 = \infty$
 karena nilai $M_{[1,3]} > M_{[1,2]} + M_{[2,3]}$ maka nilai $M_{[1,3]}$ berubah
 atau diganti menjadi 3.8.

4. Iterasi $k = 3$

Representasi matriks $M^{(3)}$ iterasi $k = 3$ dapat dilihat pada **lampiran 6**.

Selanjutnya, untuk setiap elemen matriks M dicek apakah $M_{[i,j]} > M_{[i,k]} + M_{[k,j]}$. Jika ya, maka $M_{[i,j]}$ diganti dengan $M_{[i,k]} + M_{[k,j]}$

Beberapa hasil pengecekan pada iterasi $k = 3$ adalah sebagai berikut :

- a. $M_{[1,3]} = \infty$ sedangkan $M_{[1,2]} + M_{[2,3]} = 0.8 + 3 = \infty$
 karena nilai $M_{[1,3]} > M_{[1,2]} + M_{[2,3]}$ maka nilai $M_{[1,3]}$ berubah atau
 diganti menjadi 3.8.

b. $M_{[4,1]} = \infty$ sedangkan $M_{[4,3]} + M_{[3,1]} = 7.3 + 3.8 = \infty$

karena nilai $M_{[4,1]} > M_{[4,3]} + M_{[3,1]}$ maka nilai $M_{[4,1]}$ berubah atau diubah menjadi 11.1.

c. $M_{[14,1]} = \infty$ sedangkan $M_{[14,3]} + M_{[3,1]} = 0.8 + 3.8 = 4.6$

karena nilai $M_{[14,1]} > M_{[14,3]} + M_{[3,1]}$ maka nilai $M_{[14,1]}$ berubah atau diubah menjadi 4.6.

d. $M_{[4,2]} = \infty$ sedangkan $M_{[4,3]} + M_{[3,2]} = 7.3 + 3 = 10.3$

karena nilai $M_{[4,2]} > M_{[4,3]} + M_{[3,2]}$ maka nilai $M_{[4,2]}$ berubah atau diubah menjadi 10.3.

e. $M_{[2,14]} = \infty$ sedangkan $M_{[2,3]} + M_{[3,14]} = 3 + 0.8 = 3.8$

karena nilai $M_{[2,14]} > M_{[2,3]} + M_{[3,14]}$ maka nilai $M_{[2,14]}$ berubah atau diubah menjadi 3.8.

f. $M_{[14,4]} = \infty$ sedangkan $M_{[14,3]} + M_{[3,4]} = 0.8 + 7.3 = 8.1$

karena nilai $M_{[14,4]} > M_{[14,3]} + M_{[3,4]}$ maka nilai $M_{[14,4]}$ berubah atau diubah menjadi 8.1.

5. Iterasi $k = 23$

Representasi matriks $M^{(23)}$ iterasi $k = 23$ dapat dilihat **lampiran 6**.

Matriks ini kemudian akan dijadikan tabel acuan dalam mencari jarak terpendek saat menentukan rute terpendek.

4.4 Penentuan Rute Terpendek

Setelah melakukan perhitungan sampai iterasi terakhir yaitu iterasi ke $k = 23$ kemudian matriks tersebut akan direpresentasikan ke dalam tabel (dapat dilihat

pada **lampiran 7**). Baris pada tabel tersebut berisikan titik awal, kolom pada tabel tersebut berisikan titik tujuan dan setiap elemen pada tabel tersebut berisikan sisi atau jarak terpendeknya. Berikut merupakan percobaan untuk menentukan rute terpendek dengan tahapan:

1. Penentuan Titik Awal dan Titik Tujuan

Penentuan titik awal dapat dilakukan sesuai yang diinginkan dengan memperhatikan kode titik dalam tabel 4.2 yang berisikan data titik persimpangan dan objek tempat umum di sepanjang jalan titik rawan kecelakaan.

Sebagai percobaan pertama, ditentukan titik awal adalah titik A (Persimpangan Jl. Sruwen-Karanggede) ke titik 1 dan 2 karena dari arah utara dua rumah sakit ini yang paling dekat dengan titik A. Percobaan kedua, ditentukan titik awal adalah titik B (Persimpangan Jl. Raya Salatiga-Solo) ke titik 1 dan 2 karena dari arah utara dua rumah sakit ini yang paling dekat dengan titik B. Percobaan ketiga, ditentukan titik awal adalah titik C (Tugu Lilin) ke titik 1 dan 2 karena dari arah utara dua rumah sakit ini yang paling dekat dengan titik C. Percobaan keempat, ditentukan titik awal adalah D (Terminal Bus Boyolali) ke titik 1 dan 2 karena dari arah selatan dua rumah sakit ini yang paling dekat dengan titik D. Percobaan kelima, ditentukan titik awal adalah E (Warung Soto Hj. Fatimah Pandanarang) ke titik 3 dan 6 karena dua rumah sakit tersebut yang terdekat dari titik E. Percobaan keenam, ditentukan titik awal F (Alun-alun Lor Boyolali) ke titik 6 dan 7 karena dua rumah sakit tersebut yang terdekat dari titik F.

2. Penentuan Rute

Percobaan pada poin 1 diatas kemudian dilakukan penentuan rute dari titik awal menuju titik tujuan. Untuk menentukan titik mana dengan jarak terpendek yang akan dikunjungi terlebih dahulu, maka akan digunakan algoritma Floyd-Warshall yang sudah dilakukan perhitungan di subbab 4.3. Berikut penentuan rute berdasarkan percobaan pada poin 1.

a. Penentuan rute titik A ke titik 1

Titik A adalah titik awal pada percobaan dan titik 1 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 4.6 km dengan melewati titik $A \rightarrow B \rightarrow C \rightarrow 1$.

b. Penentuan rute A ke titik 2

Titik A adalah titik awal pada percobaan dan titik 2 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 14.2 km dengan melewati titik $A \rightarrow B \rightarrow C \rightarrow D \rightarrow 2$.

c. Penentuan rute B ke titik 1

Titik B adalah titik awal pada percobaan dan titik 1 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 3.8 km dengan melewati titik $B \rightarrow C \rightarrow 1$.

d. Penentuan rute B ke titik 2

Titik B adalah titik awal pada percobaan dan titik 2 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 13.4 km dengan melewati titik $B \rightarrow C \rightarrow D \rightarrow 2$.

e. Penentuan rute C ke titik 1

Titik B adalah titik awal pada percobaan dan titik 1 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 0.8 km dengan melewati titik $C \rightarrow 1$.

f. Penentuan rute C ke titik 2

Titik B adalah titik awal pada percobaan dan titik 2 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 10.4 km dengan melewati titik $C \rightarrow D \rightarrow 2$.

g. Penentuan rute D ke titik 2

Titik B adalah titik awal pada percobaan dan titik 2 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 3.1 km dengan melewati titik $D \rightarrow 2$.

h. Penentuan rute D ke titik 3

Titik B adalah titik awal pada percobaan dan titik 2 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 4.7 km dengan melewati titik $D \rightarrow E \rightarrow 3$.

i. Penentuan rute E ke titik 3

Titik B adalah titik awal pada percobaan dan titik 3 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 1.3 km dengan melewati titik $E \rightarrow 3$.

j. Penentuan rute E ke titik 6

Titik B adalah titik awal pada percobaan dan titik merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 4.7 km dengan melewati titik $E \rightarrow F \rightarrow 6$.

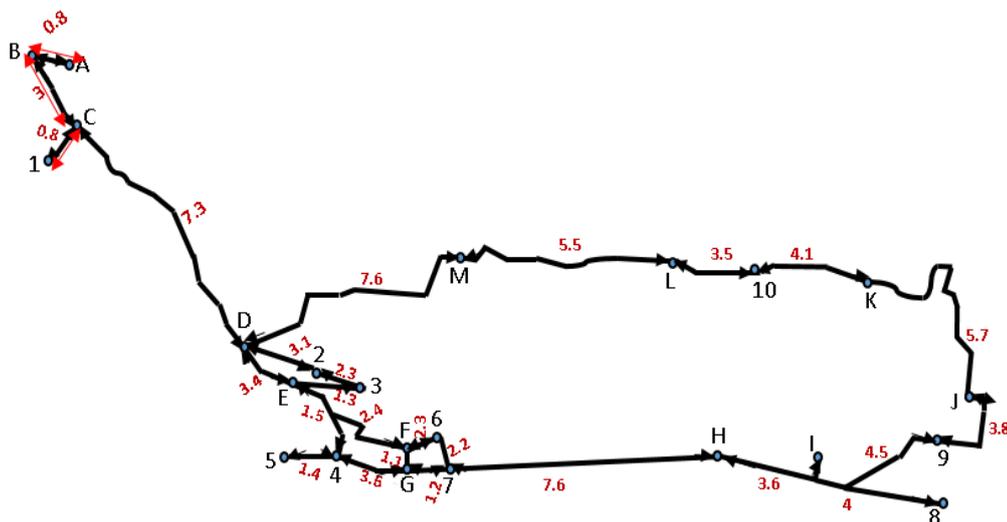
k. Penentuan rute F ke titik 6

Titik B adalah titik awal pada percobaan dan titik 6 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 2.3 km dengan melewati titik $F \rightarrow 6$.

1. Penentuan rute F ke titik 7

Titik B adalah titik awal pada percobaan dan titik 7 merupakan titik tujuan, untuk melihat jarak terpendeknya dapat dilihat pada elemen tabel **lampiran 7** dengan baris sebagai titik awal dan kolom sebagai titik tujuan. Kemudian ditemukan jarak terpendek 2.3 km dengan melewati titik $F \rightarrow G \rightarrow 7$.

Berikut merupakan hasil percobaan rute terpendek dari titik A ke titik 1 sebagai berikut:



Gambar 4.5 Hasil Percobaan Rute A ke 1

4.5 Integrasi Al-Quran Dan Hadits Terkait Kecelakaan Di Boyolali

Kecelakaan lalu lintas yang terjadi di Boyolali, Jawa Tengah adalah suatu musibah yang dialami oleh manusia. Musibah adalah takdir dari Allah *subhanahu wata'ala* yang diberikan kepada hambanya sebagai ujian kesabaran sekaligus memberikan peringatan bahwa Allah Maha Kuasa dengan segala yang terjadi di

bumi maupun di langit. Sesuai yang tertuang dalam firman Allah *subhanahu wata'ala* Al-Qur'an Surah Al-Baqarah ayat 156 yang artinya:

“(Yaitu) orang-orang yang apabila ditimpa musibah, mereka berkata ‘innaalillahi wa inna ilaihi roji’un’ (sesungguhnya kami milik Allah dan kepada-Nyalah kami kembali) (Q.S Al-Baqarah:156).

Kata musibah berasal dari Bahasa arab *‘ashaba-yashibu* dalam kamus Bahasa Arab *al-Munawwir* artinya bencana atau malapetaka. Sementara pada kamus Bahasa Indonesia artinya dengan kejadian menyedihkan yang menimpa, atau malapetaka, atau bencana. Musibah dapat menimpa siapa saja dan kapan saja. Sehingga perlu dilakukan tata cara untuk menghadapi suatu musibah agar manusia tidak salah dalam menanggapi hal tersebut (Hamka, 2015).

Terdapat tips yang dapat dilakukan dalam menghadapi musibah. Pertama, menghadapi musibah dengan *muhasabah* atau introspeksi diri terkait dengan yang telah dikerjakan selama di dunia. Kejadian yang tidak menyenangkan, seperti kecelakaan lalu lintas dapat terjadi. Sebagai manusia harus dihadapi dengan lapang dada, sabar, dan mengevaluasi diri sendiri. Musibah tersebut akan menjadikan dosa-dosa manusia diampuni jika menerima musibah tersebut tanpa mengeluh, sabar, dan tidak menyalahkan keadaan karena diberikan suatu musibah.

Kedua, menghadapi musibah dapat dilakukan dengan berpikir positif. Allah memberikan suatu musibah bukan tanpa alasan, melainkan suatu ujian bagi manusia. Jika ujian tersebut dapat dijalani dengan baik maka Allah akan memberikan sebuah ganjaran baik dihapus dosa-dosanya dan diangkat derajatnya jika menghadapinya dengan sabar. Kesabaran tersebut menjadikan keimanan seseorang semakin kuat karena musibah yang dikirimkan kepada hambanya bukan

sebagai suatu yang dianggap menyusahkan, melainkan tetap berbaik sangka kepada Allah.

Ketiga adalah tidak mengeluh dan mencela dengan musibah yang diberikan Allah *subhanahu-wata'ala* kepada kita. Menjalani suatu musibah tidaklah mudah. Akan tetapi, dengan tetap semangat dan tawakkal untuk mencari solusi tanpa mengeluh maka Allah akan memberikan kesembuhan bagi yang tertimpa musibah kecelakaan lalu lintas.

Suatu musibah tersebut dapat terjadi dengan seizin Allah. Musibah tersebut diberikan bagi yang dikehendaki-Nya. Maka tetaplah meminta petunjuk kepada Allah agar senantiasa diberikan suatu jalan terbaik. Sesuai firman Allah *subhanahu-wata'ala* dalam Al-Qur'an ayat 11 yang artinya:

“Tidak ada suatu musibah pun yang menimpa seseorang kecuali dengan izin Allah; dan barang siapa yang beriman kepada Allah niscaya Dia akan memberikan petunjuk kepada hatinya. Dan Allah Maha Mengetahui segala sesuatu” (Q.S At-Taghabun:11).

Allah senantiasa memberikan jalan keluar atas kesulitan yang menimpa manusia. Kesulitan tersebut misalnya musibah. Jika mohon petunjuk dan senantiasa berdoa dan tawakal maka Allah akan memberikan kelapangan dan kekuatan dalam menghadapi musibah tersebut (Hamka, 2015).

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, dapat disimpulkan bahwa penerapan algoritma Floyd-Warshall pada evakuasi korban kecelakaan di Boyolali dapat dilakukan dan menghasilkan matriks acuan. Matriks acuan ini berisi elemen jarak terpendek antar titik yang digunakan sebagai acuan untuk menentukan rute terpendek dari satu jalan rawan kecelakaan menuju rumah sakit terdekat dengan cara melihat baris sebagai titik awal dan kolom sebagai titik tujuan dan sebaliknya.

5.2 Saran Untuk Penelitian Selanjutnya

Penelitian selanjutnya disarankan untuk membandingkan tingkat keefektifan menggunakan Algoritma Floyd-Warshall dengan algoritma lainnya, untuk mengetahui algoritma manakah yang lebih efektif dalam menentukan rute terpendek.

DAFTAR PUSTAKA

- A, A., & Salehipour. (2018). Speeding Up The Floyd–Warshall Algorithm For The Cycled Shortest Path Problem. *Computer Informatics*, 31.
- Agung, J. O., Efendi, T., & Agung, H. (2018). Analisis Perbandingan Algoritma Floyd-Warshall dengan Algoritma Bellman-Ford Dalam Pencarian Rute Terpendek Menuju Museum di Jakarta. *Kalbi Scientia Jurnal Sains dan Teknologi*, 5 no 1, 4-5.
- Anton, H. (2000). *Dasar-dasar Aljabar Linier Edisi 7 Jilid 2*. Batam: Interaksara.
- Arfananda, M., Nasution, S., & Setianingsih, C. (2020). A Selection of Bandung City Travel Route Using The Floyd-Warshall Algorithm. *The International Journal of Integrated Engineering*, g Vol. 12 No. 7 (2020) p. 90-97, 4-5. Diambil kembali dari <http://penerbit.uthm.edu.my/ojs/index.php/ijie>
- Arumsari, N. D., Nugraha, A. L., & Awaluddin, M. (2016). Pemodelan Daerah Rawan Kecelakaan dengan Menggunakan Cluster Analysis. *Jurnal Geodesi Undip*, 181.
- Cantona, A., Fauziah, & Winarsih. (2020). Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta. *Teknologi dan Manajemen Informatika*, 6, 72-74.
- Cormen, T. (2003). Introduction To Algorithms Edition 2nd Edition. *The MIT Press Cambridge*, 5.
- Dermawan, T. (2019). Comparison of Djikstra dan Floyd-Wharshall Algorithm to Determine the Best Route of Train. *Internasional Journal on Informatics for Development*, 7 no 2, 9.
- Enggarsari, U. (2017). Kajian Terhadap Faktor-Faktor Penyebab Kecelakaan Lalu Lintas Dalam Upaya Perbaikan Pencegahan Kecelakaan Lalu Lintas. *Perspektif*, 239.
- Faizah, N. (2022). *Ternyata Ini Ruas Jalan Paling Rawan Kecelakaan di Boyolali*. Boyolali: Solopos.com. Dipetik Maret 23, 2023, dari <https://www.solopos.com/ternyata-ini-ruas-jalan-paling-rawan-kecelakaan-di-boyolali-1321432>
- Hamka. (2015). *Tafsir Al-Azhar*. Jakarta: Gema Insani.
- J, M. (2020). *Memprihatinkan, Jumlah Kecelakaan di Boyolali Meningkat 101 Persen dari Tahun 2020*. Surakarta: Solo Suara. Dipetik September 13, 2021, dari <https://solo.suaramerdeka.com/>.
- Juhari. (2021). *Matematika Diskrit*. Banyumas: CV. Pena Persada.
- Komarullah, Y. (2013). Penerapan Metode Graf dan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek. *E-journal UIN Jakarta*, 22-01-2, 8
- Maulana, G. (2017). *Pembelajaran Dasar Algoritma dan Pemrograman*

Menggunakan El-Goritma Berbasis Web. Jakarta: Teknik Mesin.

- Mohammad, H. Y., A., S. R., & S., P. (2018). Implementation Flyod-Warshall Algorithm For The Shortest Path of Garage. *International Journal of Innovative Science and Research Technology*, 3 no 2, 5-6.
- Munir, R. (2012). *Buku Teks Ilmu Komputer Matematika Diskrit* (ketiga ed.). Bandung: Informatika.
- Murdowo, J. (2022, Januari Rabu). *Memprihatinkan. Jumlah Kecelakaan di Boyolali Meningkat 101 Persen dari Tahun 2020*. Diambil kembali dari Suara Merdeka Solo: <https://solo.suaramerdeka.com/>
- Pandey, V., Yadav, S., & Arora. (2016). Retiming Technique for Clock Period Minimization using Shortest Path Algorithm. *International Conference on Computing, Communication and Automation (ICCCA)*, 1418-1423.
- Saputra, A. D. (2017). Studi Tingkat Kecelakaan Lalu Lintas Jalan di Indonesia Berdasarkan Data KNKT (Komite Nasional Keselamatan Transportasi) dari Tahun 2007-2016. *Warta Penelitian Perhubungan*, 180.
- Shahid, M., Khawar, H., & Kamal, A. M. (2021). The Mathematical Model for searching the Shortest Route for Tuberculosis Patients with the help of Dijkstra's Algorithm. *Sukkur IBA of Computing and Mathematical Sciences*, 5, 44.
- Siswanto. (2006). *Opertions Research Jilid 1*. Jakarta: Erlangga.
- Usman, S., Wiranto, I., & Nurwan. (2022). Aplikasi Algoritma Floyd-Warshall untuk Mengoptimalkan Distribusi Listrik di PLN Kota Gorontalo. *Research in the Mathematical and Natural Sciences*, 1 No 1, 8.
- Yudha, M., Rahmani, A., & Purwani, S. (2022). Finding Shortest Path at Padjadjaran University Using Floyd-Warshall Algorithm. *World Scientific News An International Scientific Journal*, 7.

LAMPIRAN

Lampiran 1 Script Program Algoritma Floyd-Warshall Dengan Matlab

```
clc
clear all
close all

%Program algoritma Floyd-Warshall

matriks_awal=[0 0.8 inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf; 0.8 0
3 inf inf
inf inf inf inf inf inf inf inf; inf 3 0 7.3 inf inf inf
inf inf inf inf inf inf 0.8 inf inf inf inf inf inf inf
inf inf; inf inf 7.3 0 3.4 inf inf inf inf inf inf inf
7.6 inf 3.1 inf inf inf inf inf inf inf inf; inf inf
inf 3.4 0 2.4 inf inf inf inf inf inf inf inf inf 1.3
1.5 inf inf inf inf inf inf inf; inf inf inf inf 2.4 0 1.1
inf 2.3 inf inf
inf inf; inf inf inf inf inf 1.1 0 inf inf inf inf inf
inf inf inf inf 3.6 inf inf 1.2 inf inf inf; inf inf
inf inf inf inf inf 0 3.6 inf inf inf inf inf inf inf
inf inf inf 7.6 inf inf inf; inf inf inf inf inf inf inf
inf 3.6 0 inf 4
4.5 inf; inf 0 5.7 inf
inf inf inf inf inf inf inf inf inf inf 3.8 inf; inf inf
inf inf inf inf inf inf inf 5.7 0 inf inf inf inf inf
inf inf inf inf inf inf 4.1; inf inf inf inf inf inf inf
inf inf inf inf inf 0 5.5 inf inf inf inf inf inf inf
inf inf 3.5; inf inf inf 7.6 inf inf inf inf inf inf inf
inf 5.5 0 inf inf; inf
inf 0.8 inf 0 inf
inf inf inf inf inf inf inf inf; inf inf inf 3.1 inf
inf inf inf inf inf inf inf inf inf inf 0 2.3 inf inf inf
inf inf inf inf; inf inf inf inf 1.3 inf inf inf inf
inf inf inf inf inf 2.3 0 inf inf inf inf inf inf inf;
inf inf inf inf 1.5 inf 3.6 inf inf inf inf inf inf inf
inf inf 0 1.4 inf inf inf inf inf; inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf 1.4 0 inf
inf inf inf inf; inf inf inf inf inf inf 2.3 inf inf inf
inf inf inf inf inf inf inf inf inf inf 0 2.2 inf inf inf;
inf inf inf inf inf inf inf inf 7.6 inf inf inf inf inf
inf inf inf inf 2.2 0 inf inf inf; inf inf inf inf inf
inf inf inf 4 inf inf inf inf inf inf inf inf inf inf
inf 0 inf inf; inf inf inf inf inf inf inf inf inf 4.5 3.8
inf 0 inf; inf
inf inf inf inf inf inf inf inf inf inf 4.1 3.5 inf inf inf
inf inf inf inf inf inf inf 0]
```

```

[baris,kolom] = size(matriks_awal);

for k = 1 : baris
    temp = matriks_awal;
    for i = 1 : baris
        for j = 1 : kolom
            if k == 23 && (i == 1 && j == 23)
            else
                temp (i,j) =
min(matriks_awal(i,j),matriks_awal(i,k)
matriks_awal(k,j));
            end
        end
    end
    matriks_awal = temp;
    fprintf('Iterasi ke %d adalah \40', k);
    matriks_awal
    pause;
end

```

Lampiran 2 Tabel Jarak Antar Titik Sebelum Dilakukan Perhitungan

| | A | B | C | D | E | F | G | H | I | J | K | L | M | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0.8 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 3 | 0 | 7.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 7.3 | 0 | 3.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 3.4 | 0 | 2.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.3 | 1.5 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 2.4 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 1.2 | 0 | 0 | 0 |
| H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4.5 | 0 |
| J | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.8 | 0 |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1 |
| L | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.5 |
| M | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1.5 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.4 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.2 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.2 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.5 | 3.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1 | 3.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Keterangan:

A – M = Jalan Rawan Kecelakaan
 1 – 10 = Rumah Sakit

Lampiran 3 Matriks Iterasi $k = 0$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.8 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 3 | 0 | 7.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 7.3 | 0 | 3.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 3.4 | 0 | 2.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.3 | 1.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 2.4 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 1.2 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4.5 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.5 |
| 13 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 | 0 | 0 | 0 | 5.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 1.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 1.5 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.4 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.2 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.2 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.5 | 3.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1 | 3.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Keterangan:

- 1 – 13 = Jalan Rawan Kecelakaan
 14 – 23 = Rumah Sakit

Lampiran 4 Matriks Iterasi $k = 1$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.8 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 3 | 0 | 7.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 7.3 | 0 | 3.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 3.4 | 0 | 2.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.3 | 1.5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 2.4 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 1.2 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4.5 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.8 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.5 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 1.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 1.5 | 0 | 3.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.4 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.2 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 7.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.2 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.5 | 3.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1 | 3.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Keterangan:

- 1 – 13 = Jalan Rawan Kecelakaan
 14 – 23 = Rumah Sakit

Lampiran 5 Matriks Iterasi $k = 2$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|
| 1 | 0 | 0.8 | 3.8 | | | | | | | | | | | | | | | | | | | | |
| 2 | 0.8 | 0 | 3 | | | | | | | | | | | | | | | | | | | | |
| 3 | 3.8 | 3 | 0 | 7.3 | | | | | | | | | | 0.8 | | | | | | | | | |
| 4 | | | 7.3 | 0 | 3.4 | | | | | | | | 7.6 | | 3.1 | | | | | | | | |
| 5 | | | | 3.4 | 0 | 2.4 | | | | | | | | | | 1.3 | 1.5 | | | | | | |
| 6 | | | | | 2.4 | 0 | 1.1 | | | | | | | | | | | | 2.3 | | | | |
| 7 | | | | | | 1.1 | 0 | | | | | | | | | | 3.6 | | | 1.2 | | | |
| 8 | | | | | | | 0 | 3.6 | | | | | | | | | | | | 7.6 | | | |
| 9 | | | | | | | 3.6 | 0 | | | | | | | | | | | | | 4 | 4.5 | |
| 10 | | | | | | | | | 0 | 5.7 | | | | | | | | | | | | | 3.8 |
| 11 | | | | | | | | | 5.7 | 0 | | | | | | | | | | | | | 4.1 |
| 12 | | | | | | | | | | | 0 | 5.5 | | | | | | | | | | | 3.5 |
| 13 | | | | | | | | | | | 5.5 | 0 | | | | | | | | | | | |
| 14 | | | | 0.8 | | | | | | | | | 0 | | | | | | | | | | |
| 15 | | | | | 3.1 | | | | | | | | | | 0 | 2.3 | | | | | | | |
| 16 | | | | | | 1.3 | | | | | | | | | | 2.3 | 0 | | | | | | |
| 17 | | | | | | | 3.6 | | | | | | | | | | 0 | 1.4 | | | | | |
| 18 | | | | | | | | | | | | | | | | | 1.4 | 0 | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | 0 | 2.2 | | | |
| 20 | | | | | | | | 7.6 | | | | | | | | | | | | 2.2 | 0 | | |
| 21 | | | | | | | | | 4 | | | | | | | | | | | | 0 | | |
| 22 | | | | | | | | | 4.5 | 3.8 | | | | | | | | | | | | 0 | |
| 23 | | | | | | | | | | | 4.1 | 3.5 | | | | | | | | | | | 0 |

Keterangan:

1 – 13 || Jalan Rawan Kecelakaan
 14 – 23 || Rumah Sakit

Lampiran 6 Matriks Iterasi $k = 3$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
|----|------|------|-----|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|-----|
| 1 | 0 | 0.8 | 3.8 | 11.1 | | | | | | | | | | 4.6 | | | | | | | | | | |
| 2 | 0.8 | 0 | 3 | 10.3 | | | | | | | | | | 3.8 | | | | | | | | | | |
| 3 | 3.8 | 3 | 0 | 7.3 | | | | | | | | | | 0.8 | | | | | | | | | | |
| 4 | 11.1 | 10.3 | 7.3 | 0 | 3.4 | | | | | | | | 7.6 | 8.1 | 3.1 | | | | | | | | | |
| 5 | | | | 3.4 | 0 | 2.4 | | | | | | | | | 1.3 | 1.5 | | | | | | | | |
| 6 | | | | | 2.4 | 0 | 1.1 | | | | | | | | | | | | 2.3 | | | | | |
| 7 | | | | | | 1.1 | 0 | | | | | | | | | | 3.6 | | | 1.2 | | | | |
| 8 | | | | | | | | 0 | 3.6 | | | | | | | | | | | 7.6 | | | | |
| 9 | | | | | | | | 3.6 | 0 | | | | | | | | | | | | 4 | 4.5 | | |
| 10 | | | | | | | | | | 0 | 5.7 | | | | | | | | | | | | 3.8 | |
| 11 | | | | | | | | | | 5.7 | 0 | | | | | | | | | | | | | 4.1 |
| 12 | | | | | | | | | | | | 0 | 5.5 | | | | | | | | | | | 3.5 |
| 13 | | | | | | | | | | | | 5.5 | 0 | | | | | | | | | | | |
| 14 | 4.6 | 3.8 | 0.8 | 8.1 | | | | | | | | | | 0 | | | | | | | | | | |
| 15 | | | | 3.1 | | | | | | | | | | | 0 | 2.3 | | | | | | | | |
| 16 | | | | | 1.3 | | | | | | | | | | 2.3 | 0 | | | | | | | | |
| 17 | | | | | 1.5 | | 3.6 | | | | | | | | | | 0 | 1.4 | | | | | | |
| 18 | | | | | | | | | | | | | | | | | 1.4 | 0 | | | | | | |
| 19 | | | | | | 2.3 | | | | | | | | | | | | | 0 | 2.2 | | | | |
| 20 | | | | | | | | 7.6 | | | | | | | | | | | 2.2 | 0 | | | | |
| 21 | | | | | | | | | 4 | | | | | | | | | | | | 0 | | | |
| 22 | | | | | | | | | 4.5 | 3.8 | | | | | | | | | | | | 0 | | |
| 23 | | | | | | | | | | | 4.1 | 3.5 | | | | | | | | | | | | 0 |

Keterangan:

- 1 – 13 = Jalan Rawan Kecelakaan
 14 – 23 = Rumah Sakit

Lampiran 7 Matriks Iterasi $k = 23$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 0 | 0.8 | 3.8 | 11.1 | 14.5 | 16.9 | 18 | 26.8 | 30.4 | 37.5 | 31.8 | 24.2 | 18.7 | 4.6 | 14.2 | 15.8 | 16 | 17.4 | 19.2 | 19.2 | 34.4 | 34.9 | 27.7 |
| 2 | 0.8 | 0 | 3 | 10.3 | 13.7 | 16.1 | 17.2 | 26 | 29.6 | 36.7 | 31 | 23.4 | 17.9 | 3.8 | 13.4 | 15 | 15.2 | 16.6 | 18.4 | 18.4 | 33.6 | 34.1 | 26.9 |
| 3 | 3.8 | 3 | 0 | 7.3 | 10.7 | 13.1 | 14.2 | 23 | 26.6 | 33.7 | 28 | 20.4 | 14.9 | 0.8 | 10.4 | 12 | 12.2 | 13.6 | 15.4 | 15.4 | 30.6 | 31.1 | 23.9 |
| 4 | 11.1 | 10.3 | 7.3 | 0 | 3.4 | 5.8 | 6.9 | 15.7 | 19.3 | 26.4 | 20.7 | 13.1 | 7.6 | 8.1 | 3.1 | 4.7 | 4.9 | 6.3 | 8.1 | 8.1 | 23.3 | 23.8 | 16.6 |
| 5 | 14.5 | 13.7 | 10.7 | 3.4 | 0 | 2.4 | 3.5 | 12.3 | 15.9 | 24.2 | 24.1 | 16.5 | 11 | 11.5 | 3.6 | 1.3 | 1.5 | 2.9 | 4.7 | 4.7 | 19.9 | 20.4 | 20 |
| 6 | 16.9 | 16.1 | 13.1 | 5.8 | 2.4 | 0 | 1.1 | 9.9 | 13.5 | 21.8 | 26.5 | 18.9 | 13.4 | 13.9 | 6 | 3.7 | 3.9 | 5.3 | 2.3 | 2.3 | 17.5 | 18 | 22.4 |
| 7 | 18 | 17.2 | 14.2 | 6.9 | 3.5 | 1.1 | 0 | 8.8 | 12.4 | 20.7 | 26.4 | 20 | 14.5 | 15 | 7.1 | 4.8 | 3.6 | 5 | 3.4 | 1.2 | 16.4 | 16.9 | 23.5 |
| 8 | 29 | 28.2 | 25.2 | 17.9 | 14.5 | 12.1 | 13.2 | 0 | 3.6 | 11.9 | 17.6 | 25.2 | 25.5 | 26 | 18.1 | 15.8 | 16 | 17.4 | 9.8 | 7.6 | 7.6 | 8.1 | 21.7 |
| 9 | 32.6 | 31.8 | 28.8 | 21.5 | 18.1 | 15.7 | 16.8 | 3.6 | 0 | 8.3 | 14 | 21.6 | 27.1 | 29.6 | 21.7 | 19.4 | 19.6 | 21 | 13.4 | 11.2 | 4 | 4.5 | 18.1 |
| 10 | 37.5 | 36.7 | 33.7 | 26.4 | 26.4 | 24 | 25.1 | 11.9 | 8.3 | 0 | 5.7 | 13.3 | 18 | 34.5 | 29.5 | 27.7 | 27.9 | 29.3 | 21.7 | 19.5 | 12.3 | 3.8 | 9.8 |
| 11 | 31.8 | 31 | 28 | 20.7 | 24.1 | 26.5 | 27.6 | 17.6 | 14 | 5.7 | 0 | 7.6 | 13.1 | 28.8 | 23.8 | 25.4 | 25.6 | 27 | 27.4 | 25.2 | 18 | 9.5 | 4.1 |
| 12 | 24.2 | 23.4 | 20.4 | 13.1 | 16.5 | 18.9 | 20 | 25.2 | 21.6 | 13.3 | 7.6 | 0 | 5.5 | 21.2 | 16.2 | 17.8 | 18 | 19.4 | 21.2 | 21.2 | 25.6 | 17.1 | 3.5 |
| 13 | 18.7 | 17.9 | 14.9 | 7.6 | 11 | 13.4 | 14.5 | 23.3 | 26.9 | 18.8 | 13.1 | 5.5 | 0 | 15.7 | 10.7 | 12.3 | 12.5 | 13.9 | 15.7 | 15.7 | 30.9 | 22.6 | 9 |
| 14 | 4.6 | 3.8 | 0.8 | 8.1 | 11.5 | 13.9 | 15 | 23.8 | 27.4 | 34.5 | 28.8 | 21.2 | 15.7 | 0 | 11.2 | 12.8 | 13 | 14.4 | 16.2 | 16.2 | 31.4 | 31.9 | 24.7 |
| 15 | 14.2 | 13.4 | 10.4 | 3.1 | 3.6 | 6 | 7.1 | 15.9 | 19.5 | 27.8 | 23.8 | 16.2 | 10.7 | 11.2 | 0 | 2.3 | 5.1 | 6.5 | 8.3 | 8.3 | 23.5 | 24 | 19.7 |
| 16 | 15.8 | 15 | 12 | 4.7 | 1.3 | 3.7 | 4.8 | 13.6 | 17.2 | 25.5 | 25.4 | 17.8 | 12.3 | 12.8 | 2.3 | 0 | 2.8 | 4.2 | 6 | 6 | 21.2 | 21.7 | 21.3 |
| 17 | 16 | 15.2 | 12.2 | 4.9 | 1.5 | 3.9 | 3.6 | 12.4 | 16 | 24.3 | 25.6 | 18 | 12.5 | 13 | 5.1 | 2.8 | 0 | 1.4 | 6.2 | 4.8 | 20 | 20.5 | 21.5 |
| 18 | 17.4 | 16.6 | 13.6 | 6.3 | 2.9 | 5.3 | 5 | 13.8 | 17.4 | 25.7 | 27 | 19.4 | 13.9 | 14.4 | 6.5 | 4.2 | 1.4 | 0 | 7.6 | 6.2 | 21.4 | 21.9 | 22.9 |
| 19 | 19.2 | 18.4 | 15.4 | 8.1 | 4.7 | 2.3 | 3.4 | 9.8 | 13.4 | 21.7 | 27.4 | 21.2 | 15.7 | 16.2 | 8.3 | 6 | 6.2 | 7.6 | 0 | 2.2 | 17.4 | 17.9 | 24.7 |
| 20 | 21.4 | 20.6 | 17.6 | 10.3 | 6.9 | 4.5 | 5.6 | 7.6 | 11.2 | 19.5 | 25.2 | 23.4 | 17.9 | 18.4 | 10.5 | 8.2 | 8.4 | 9.8 | 2.2 | 0 | 15.2 | 15.7 | 26.9 |
| 21 | 36.6 | 35.8 | 32.8 | 25.5 | 22.1 | 19.7 | 20.8 | 8.1 | 4 | 12.3 | 18 | 25.6 | 30.1 | 33.6 | 25.7 | 23.4 | 23.6 | 25 | 17.4 | 15.2 | 0 | 8.5 | 22.1 |
| 22 | 37.1 | 36.3 | 33.3 | 26 | 22.6 | 20.2 | 21.3 | 21.7 | 4.5 | 3.8 | 9.5 | 17.1 | 22.6 | 34.1 | 26.2 | 23.9 | 24.1 | 25.5 | 17.9 | 15.7 | 8.5 | 0 | 13.6 |
| 23 | 27.7 | 26.9 | 23.9 | 16.6 | 20 | 22.4 | 23.5 | 21.7 | 18.1 | 9.8 | 4.1 | 3.5 | 9 | 24.7 | 19.7 | 21.3 | 21.5 | 22.9 | 24.7 | 24.7 | 22.1 | 13.6 | 0 |

Keterangan:

- 1 – 13 = Jalan Rawan Kecelakaan
 14 – 23 = Rumah Sakit

Lampiran 8 Jarak Antar Titik Setelah Dilakukan Perhitungan

| | A | B | C | D | E | F | G | H | I | J | K | L | M | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| A | 0 | 0.8 | 3.8 | 11.1 | 14.5 | 16.9 | 18 | 26.8 | 30.4 | 37.5 | 31.8 | 24.2 | 18.7 | 4.6 | 14.2 | 15.8 | 16 | 17.4 | 19.2 | 19.2 | 34.4 | 34.9 | 27.7 |
| B | 0.8 | 0 | 3 | 10.3 | 13.7 | 16.1 | 17.2 | 26 | 29.6 | 36.7 | 31 | 23.4 | 17.9 | 3.8 | 13.4 | 15 | 15.2 | 16.6 | 18.4 | 18.4 | 33.6 | 34.1 | 26.9 |
| C | 3.8 | 3 | 0 | 7.3 | 10.7 | 13.1 | 14.2 | 23 | 26.6 | 33.7 | 28 | 20.4 | 14.9 | 0.8 | 10.4 | 12 | 12.2 | 13.6 | 15.4 | 15.4 | 30.6 | 31.1 | 23.9 |
| D | 11.1 | 10.3 | 7.3 | 0 | 3.4 | 5.8 | 6.9 | 15.7 | 19.3 | 26.4 | 20.7 | 13.1 | 7.6 | 8.1 | 3.1 | 4.7 | 4.9 | 6.3 | 8.1 | 8.1 | 23.3 | 23.8 | 16.6 |
| E | 14.5 | 13.7 | 10.7 | 3.4 | 0 | 2.4 | 3.5 | 12.3 | 15.9 | 24.2 | 24.1 | 16.5 | 11 | 11.5 | 3.6 | 1.3 | 1.5 | 2.9 | 4.7 | 4.7 | 19.9 | 20.4 | 20 |
| F | 16.9 | 16.1 | 13.1 | 5.8 | 2.4 | 0 | 1.1 | 9.9 | 13.5 | 21.8 | 26.5 | 18.9 | 13.4 | 13.9 | 6 | 3.7 | 3.9 | 5.3 | 2.3 | 2.3 | 17.5 | 18 | 22.4 |
| G | 18 | 17.2 | 14.2 | 6.9 | 3.5 | 1.1 | 0 | 8.8 | 12.4 | 20.7 | 26.4 | 20 | 14.5 | 15 | 7.1 | 4.8 | 3.6 | 5 | 3.4 | 1.2 | 16.4 | 16.9 | 23.5 |
| H | 29 | 28.2 | 25.2 | 17.9 | 14.5 | 12.1 | 13.2 | 0 | 3.6 | 11.9 | 17.6 | 25.2 | 25.5 | 26 | 18.1 | 15.8 | 16 | 17.4 | 9.8 | 7.6 | 7.6 | 8.1 | 21.7 |
| I | 32.6 | 31.8 | 28.8 | 21.5 | 18.1 | 15.7 | 16.8 | 3.6 | 0 | 8.3 | 14 | 21.6 | 27.1 | 29.6 | 21.7 | 19.4 | 19.6 | 21 | 13.4 | 11.2 | 4 | 4.5 | 18.1 |
| J | 37.5 | 36.7 | 33.7 | 26.4 | 26.4 | 24 | 25.1 | 11.9 | 8.3 | 0 | 5.7 | 13.3 | 18 | 34.5 | 29.5 | 27.7 | 27.9 | 29.3 | 21.7 | 19.5 | 12.3 | 3.8 | 9.8 |
| K | 31.8 | 31 | 28 | 20.7 | 24.1 | 26.5 | 27.6 | 17.6 | 14 | 5.7 | 0 | 7.6 | 13.1 | 28.8 | 23.8 | 25.4 | 25.6 | 27 | 27.4 | 25.2 | 18 | 9.5 | 4.1 |
| L | 24.2 | 23.4 | 20.4 | 13.1 | 16.5 | 18.9 | 20 | 25.2 | 21.6 | 13.3 | 7.6 | 0 | 5.5 | 21.2 | 16.2 | 17.8 | 18 | 19.4 | 21.2 | 21.2 | 25.6 | 17.1 | 3.5 |
| M | 18.7 | 17.9 | 14.9 | 7.6 | 11 | 13.4 | 14.5 | 23.3 | 26.9 | 18.8 | 13.1 | 5.5 | 0 | 15.7 | 10.7 | 12.3 | 12.5 | 13.9 | 15.7 | 15.7 | 30.9 | 22.6 | 9 |
| 1 | 4.6 | 3.8 | 0.8 | 8.1 | 11.5 | 13.9 | 15 | 23.8 | 27.4 | 34.5 | 28.8 | 21.2 | 15.7 | 0 | 11.2 | 12.8 | 13 | 14.4 | 16.2 | 16.2 | 31.4 | 31.9 | 24.7 |
| 2 | 14.2 | 13.4 | 10.4 | 3.1 | 3.6 | 6 | 7.1 | 15.9 | 19.5 | 27.8 | 23.8 | 16.2 | 10.7 | 11.2 | 0 | 2.3 | 5.1 | 6.5 | 8.3 | 8.3 | 23.5 | 24 | 19.7 |
| 3 | 15.8 | 15 | 12 | 4.7 | 1.3 | 3.7 | 4.8 | 13.6 | 17.2 | 25.5 | 25.4 | 17.8 | 12.3 | 12.8 | 2.3 | 0 | 2.8 | 4.2 | 6 | 6 | 21.2 | 21.7 | 21.3 |
| 4 | 16 | 15.2 | 12.2 | 4.9 | 1.5 | 3.9 | 3.6 | 12.4 | 16 | 24.3 | 25.6 | 18 | 12.5 | 13 | 5.1 | 2.8 | 0 | 1.4 | 6.2 | 4.8 | 20 | 20.5 | 21.5 |
| 5 | 17.4 | 16.6 | 13.6 | 6.3 | 2.9 | 5.3 | 5 | 13.8 | 17.4 | 25.7 | 27 | 19.4 | 13.9 | 14.4 | 6.5 | 4.2 | 1.4 | 0 | 7.6 | 6.2 | 21.4 | 21.9 | 22.9 |
| 6 | 19.2 | 18.4 | 15.4 | 8.1 | 4.7 | 2.3 | 3.4 | 9.8 | 13.4 | 21.7 | 27.4 | 21.2 | 15.7 | 16.2 | 8.3 | 6 | 6.2 | 7.6 | 0 | 2.2 | 17.4 | 17.9 | 24.7 |
| 7 | 21.4 | 20.6 | 17.6 | 10.3 | 6.9 | 4.5 | 5.6 | 7.6 | 11.2 | 19.5 | 25.2 | 23.4 | 17.9 | 18.4 | 10.5 | 8.2 | 8.4 | 9.8 | 2.2 | 0 | 15.2 | 15.7 | 26.9 |
| 8 | 36.6 | 36.8 | 32.8 | 25.5 | 22.1 | 19.7 | 20.8 | 8.1 | 4 | 12.3 | 18 | 25.6 | 31.1 | 33.6 | 25.7 | 23.4 | 23.6 | 25 | 17.4 | 15.2 | 0 | 8.5 | 22.1 |
| 9 | 37.1 | 36.3 | 33.3 | 26 | 22.6 | 20.2 | 21.3 | 21.7 | 4.5 | 3.8 | 9.5 | 17.1 | 22.6 | 34.1 | 26.2 | 23.9 | 24.1 | 25.5 | 17.9 | 15.7 | 8.5 | 0 | 13.6 |
| 10 | 27.7 | 26.9 | 23.9 | 16.6 | 20 | 22.4 | 23.5 | 21.7 | 18.1 | 9.8 | 4.1 | 3.5 | 9 | 24.7 | 19.7 | 21.3 | 21.5 | 22.9 | 24.7 | 24.7 | 22.1 | 13.6 | 0 |

Keterangan:

A – M = Jalan Rawan Kecelakaan
 1 – 10 = Rumah Sakit

Lampiran 9 Uji Manual

Hasil pengecekan rute dari titik A ke titik 1 (Baris 1 Kolom 14)

1. Iterasi $k = 1$
 $M_{[1,14]} = \infty$ sedangkan $M_{[1,1]} + M_{[1,14]} = \infty + \infty = \infty$
 Karena nilai $M_{[1,14]} = M_{[1,1]} + M_{[1,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai ∞
2. Iterasi $k = 2$
 $M_{[1,14]} = \infty$ sedangkan $M_{[1,2]} + M_{[2,14]} = 0.8 + \infty = \infty$
 Karena nilai $M_{[1,14]} = M_{[1,2]} + M_{[2,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai ∞
3. Iterasi $k = 3$
 $M_{[1,14]} = \infty$ sedangkan $M_{[1,3]} + M_{[3,14]} = 3.8 + 0.8 = 4.6$
 Karena nilai $M_{[1,14]} > M_{[1,3]} + M_{[3,14]}$ maka nilai $M_{[1,14]}$ berubah menjadi bernilai 4.6
4. Iterasi $k = 4$
 $M_{[1,14]} = 4.6$ sedangkan $M_{[1,4]} + M_{[4,14]} = 11.1 + 0.8 = 11.9$
 Karena nilai $M_{[1,14]} < M_{[1,4]} + M_{[4,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6
5. Iterasi $k = 5$
 $M_{[1,14]} = 4.6$ sedangkan $M_{[1,5]} + M_{[5,14]} = 14.5 + 11.5 = 26$
 Karena nilai $M_{[1,14]} < M_{[1,5]} + M_{[5,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6
6. Iterasi $k = 6$
 $M_{[1,14]} = 4.6$ sedangkan $M_{[1,6]} + M_{[6,14]} = 16.9 + 13.9 = 30.8$
 Karena nilai $M_{[1,14]} < M_{[1,6]} + M_{[6,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6
7. Iterasi $k = 7$
 $M_{[1,14]} = 4.6$ sedangkan $M_{[1,7]} + M_{[7,14]} = 18 + 15 = 23$
 Karena nilai $M_{[1,14]} < M_{[1,7]} + M_{[7,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6
8. Iterasi $k = 8$
 $M_{[1,14]} = 4.6$ sedangkan $M_{[1,8]} + M_{[8,14]} = \infty + \infty = \infty$
 Karena nilai $M_{[1,14]} < M_{[1,8]} + M_{[8,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6
9. Iterasi $k = 9$
 $M_{[1,14]} = 4.6$ sedangkan $M_{[1,9]} + M_{[9,14]} = \infty + \infty = \infty$
 Karena nilai $M_{[1,14]} < M_{[1,9]} + M_{[9,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6
10. Iterasi $k = 10$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,10]} + M_{[10,14]} = \infty + \infty = \infty$$

Karena nilai $M_{[1,14]} < M_{[1,10]} + M_{[10,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

11. Iterasi $k = 11$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,11]} + M_{[11,14]} = \infty + \infty = \infty$$

Karena nilai $M_{[1,14]} < M_{[1,11]} + M_{[11,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

12. Iterasi $k = 12$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,12]} + M_{[12,14]} = \infty + \infty = \infty$$

Karena nilai $M_{[1,14]} < M_{[1,12]} + M_{[12,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

13. Iterasi $k = 13$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,13]} + M_{[13,14]} = 18.7 + 15.7 = 34.4$$

Karena nilai $M_{[1,14]} < M_{[1,13]} + M_{[13,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

14. Iterasi $k = 15$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,15]} + M_{[15,14]} = 14.2 + 11.2 = 25.4$$

Karena nilai $M_{[1,14]} < M_{[1,15]} + M_{[15,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

15. Iterasi $k = 16$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,16]} + M_{[16,14]} = 15.8 + 12.8 = 28.6$$

Karena nilai $M_{[1,14]} < M_{[1,16]} + M_{[16,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

16. Iterasi $k = 17$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,17]} + M_{[17,14]} = 16 + 13 = 29$$

Karena nilai $M_{[1,14]} < M_{[1,17]} + M_{[17,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

17. Iterasi $k = 18$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,18]} + M_{[18,14]} = 17.4 + 14.4 = 31.8$$

Karena nilai $M_{[1,14]} < M_{[1,18]} + M_{[18,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

18. Iterasi $k = 19$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,19]} + M_{[19,14]} = 19.2 + 16.2 = 35.4$$

Karena nilai $M_{[1,14]} < M_{[1,19]} + M_{[19,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

19. Iterasi $k = 20$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,20]} + M_{[20,14]} = 18.4 + 19.2 = 37.6$$

Karena nilai $M_{[1,14]} < M_{[1,20]} + M_{[20,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

20. Iterasi $k = 21$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,21]} + M_{[21,14]} = 34.4 + 33.6 = 68$$

Karena nilai $M_{[1,14]} < M_{[1,21]} + M_{[21,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

21. Iterasi $k = 22$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,22]} + M_{[22,14]} = 34.9 + 34.1 = 69$$

Karena nilai $M_{[1,14]} < M_{[1,22]} + M_{[22,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

22. Iterasi $k = 23$

$$M_{[1,14]} = 4.6 \text{ sedangkan } M_{[1,23]} + M_{[23,14]} = 27.7 + 24.7 = 52.4$$

Karena nilai $M_{[1,14]} < M_{[1,23]} + M_{[23,14]}$ maka nilai $M_{[1,14]}$ tetap bernilai 4.6

RIWAYAT HIDUP



Fitri Nofita Sari atau kerap dipanggil Fitri dilahirkan di Kabupaten Boyolali, Jawa Tengah pada 13 September 2000, putri dari pasangan Bapak Suparman dan Ibu Suwarni. Penulis merupakan anak terakhir dari dua bersaudara. Penulis memulai pendidikan di TK Bakti II Ngesrep pada tahun 2005, lalu penulis melanjutkan pendidikan ke MI Al-Islam I kemudian penulis ke SMP N. Yaitu di SMP Negeri 2 Colomadu dan selesai pada

tahun 2016, setelah itu penulis melanjutkan di SMA N 1 Ngemplak dan lulus pada tahun 2019. Pada tahun yang sama penulis terdaftar sebagai Mahasiswa di program studi Matematika, program Strata Satu (S1) di Universitas Islam Negeri Maulana Malik Ibrahim Malang melalui jalur SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

Selama studi di Penulis aktif organisasi Himpunan Mahasiswa dan Komunitas yang ada di jurusan Matematika. Komunitas tersebut yaitu Semata atau Serambi Anak Matematika. Selain itu juga penulis aktif mengikuti beberapa pelatihan dan magang.



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

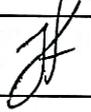
Nama : Fitri Nofita Sari
NIM : 19610004
Fakultas / Program Studi : Sains dan Teknologi / Matematika
Judul Skripsi : Penerapan Algoritma Floyd-Warshall Pada
Penentuan Jalur Terpendek Evakuasi Korban
Kecelakaan Di Boyolali Jawa Tengah
Pembimbing I : Juhari, M.Si.
Pembimbing II : Erna Herawati, M.Pd.

| No | Tanggal | Hal | Tanda Tangan |
|-----|------------------|-----------------------------------|--------------|
| 1. | 2 Januari 2023 | Konsultasi Bab I, II, dan III | 1. |
| 2. | 4 Januari 2023 | Revisi Bab I,II, dan III | 2. |
| 3. | 9 Januari 2023 | Konsultasi Kajian Agama | 3. |
| 4. | 12 Januari 2023 | Revisi Kajian Agama | 4. |
| 5. | 17 Januari 2023 | ACC untuk Seminar Proposal | 5. |
| 6. | 13 Februari 2023 | Konsultasi Bab IV dan V | 6. |
| 7. | 20 Februari 2023 | Revisi Bab IV dan V | 7. |
| 8. | 13 Maret 2023 | ACC untuk Seminar Hasil | 8. |
| 9. | 17 Mei 2023 | Revisi Bab IV dan V | 9. |
| 10. | 19 Mei 2023 | Revisi Kajian Integrasi Al-Qur'an | 10. |
| 11. | 22 Mei 2023 | ACC untuk Ujian Skripsi | 11. |
| 12. | 29 Mei 2023 | Revisi Bab III Flowchart | 12. |
| 13. | 30 Mei 2023 | Revisi Bab II Integrasi Al-Qur'an | 13. |



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

| | | | |
|-----|-------------|-----------------|---|
| 14. | 7 Juni 2023 | ACC Keseluruhan | 14.  |
|-----|-------------|-----------------|---|

Malang, 7 Juni 2023

Mengetahui,

Ketua Program Studi Matematika




Dr. Elly Susanti, M.Sc

NIP. 19741129 200012 2 005