PENERAPAN ALGORITMA DIJKSTRA DAN FLOYD-WARSHALL UNTUK MENENTUKAN RUTE TERPENDEK TEMPAT WISATA DI BATANG

Skripsi

disusun sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Program Studi Matematika

oleh
Leni Marlina
4111412038

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2017

PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, kecuali yang secara tertulis dirujuk dalam skripsi ini dan disebutkan dalam daftar pustaka. Apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, Februari 2017

Leni Marlina

4111412038

UNNES
UNIVERSITAS NEGERI SEMARANG

PENGESAHAN

Skripsi yang berjudul

Penerapan Algoritma Dijkstra dan Floyd-Warshall untuk Menentukan Rute Terpendek Tempat Wisata di Batang

disusun oleh

Leni Marlina

4111412038

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal 6 Februari 2017.

Panitia

Ketua

Prof. Dr. Zaenuri, S.E., M.Si., Akt.
196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si.
196807221993031005

Ketua Penguji

Dr. Mulyono, M.Si.
197009021997021001

Anggota Penguji/
Pembimbing I

Drs. Amin Suyitno, M.Pd.
195206041976121001

Anggota Penguji/
Pembimbing II

Drs. Mashuri, M.Si.
196708101992031003

UNNES
UNIVERSITAS NEGERI SEMARANG

MOTTO DAN PERSEMBAHAN

MOTTO

Orang-orang hebat di bidang apa pun bukan baru bekerja karena mereka terinspirasi, namun mereka menjadi terinspirasi karena mereka lebih suka bekerja, mereka tidak menyia-nyiakan waktu untuk menunggu inspirasi.

(Ernest Newman)

Kebanggaan kita yang terbesar adalah bukan tidak pernah gagal, tetapi bangkit kembali setiap kali kita jatuh.

(Confusius)

Learn from yesterday, live for today, and hope for tomorrow.

(Albert Einstein)

PERSEMBAHAN

Karya ini saya persembahkan untuk:

- Ayah dan Ibu tercinta yang telah memberikan kasih sayang, doa, dan semangat sehingga karya ini dapat terselesaikan.
- Kakak tersayang yang telah memberikan motivasi, doa, dan semangat dalam penulisan karya ini.
- Teman-teman Matematika angkatan 2012 yang telah memberikan bantuan dan semangat.
- Sahabat yang telah memberikan semangat.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyusun dan menyelesaikan skripsi yang berjudul "**Penerapan Algoritma Dijkstra dan Floyd-Warshall untuk Menentukan Rute Terpendek Tempat Wisata di Batang**".

Skripsi ini disusun untuk menyelesaikan Studi S1 untuk memperoleh gelar Sarjana Sains. Penulis menyadari bahwa skripsi ini selesai berkat bantuan, petunjuk, saran, bimbingan, dan dorongan dari berbagai pihak. Oleh karena itu perkenankanlah penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri, S.E., M.Si., Akt., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Drs. Mashuri, M.Si., Koordinator Prodi Matematika FMIPA Universitas Negeri Semarang sekaligus Dosen Pembimbing II yang dengan sabar memberikan bimbingan, petunjuk, motivasi, dan saran yang sangat bermanfaat kepada penulis selama penyusunan skripsi ini.
5. Drs. Amin Suyitno, M.Pd., Dosen Pembimbing I yang dengan sabar memberikan bimbingan, petunjuk, motivasi, dan saran yang sangat bermanfaat kepada penulis selama penyusunan skripsi ini.

UNNES
UNIVERSITAS NEGERI SEMARANG

6. Dr. Mulyono, M.Si., Dosen Penguji yang telah memberikan masukan dan saran yang membangun kepada penulis demi kesempurnaan penyusunan skripsi ini.
7. Dra. Kristina Wijayanti, M.S., Dosen Wali sekaligus sebagai inspirator dalam memberikan pencerahan dan dukungan untuk terus melangkah menyusun skripsi.
8. Seluruh Dosen Matematika yang telah membimbing dan memberikan ilmunya kepada penulis.
9. Keluarga tercinta yang selalu memberi motivasi baik moral maupun material serta doa restu dalam menyelesaikan skripsi ini.
10. Teman-teman Matematika angkatan 2012 yang telah memberikan motivasi, semangat, dan bantuannya dalam menyelesaikan skripsi ini.
11. Semua pihak yang tidak dapat penulis sebutkan satu-persatu.

Penulis berharap, semoga penelitian yang telah dilakukan ini bermanfaat bagi pembaca pada khususnya dan perkembangan pendidikan Indonesia pada umumnya.

UNNES
UNIVERSITAS NEGERI SEMARANG

Semarang, Februari 2017

Penulis

ABSTRAK

Marlina, Leni. 2016. *Penerapan Algoritma Dijkstra dan Floyd-Warshall untuk Menentukan Rute Terpendek Tempat Wisata di Batang*. Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Utama Drs. Amin Suyitno, M.Pd. dan Pembimbing Pendamping Drs. Mashuri, M.Si.

Kata kunci: Dijkstra, Floyd-Warshall, Rute Terpendek.

Fasilitas petunjuk arah menuju tempat-tempat wisata di Batang sangat minim, sehingga para wisatawan kesulitan mencari rute yang efisien menuju tempat-tempat wisata tersebut. Pencarian rute yang paling efisien ini menuntut wisatawan membuat keputusan rute mana yang diambil agar menghemat waktu dan biaya. Permasalahannya adalah bagaimana menentukan rute terpendek tempat wisata di Batang menggunakan algoritma Dijkstra dan Floyd-Warshall. Tujuan penelitian ini adalah untuk menemukan penyelesaian dari penerapan algoritma Dijkstra dan Floyd-Warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang.

Langkah-langkah dari penelitian meliputi (1) membuat graf berbobot rute tempat wisata di Batang dengan bantuan *coreldraw*, (2) menemukan penyelesaian dari penerapan algoritma Dijkstra dalam menentukan rute terpendek tempat wisata di Batang dengan bantuan *microsoft excel*, (3) menemukan penyelesaian dari penerapan algoritma Floyd-Warshall dalam menentukan rute terpendek tempat wisata di Batang dengan bantuan *microsoft excel*, (4) menentukan rute terpendek yang direkomendasikan.

Berdasarkan hasil penelitian diperoleh 27 rute terpendek di mana 25 rute adalah sama dan terdapat 2 rute yang berbeda. Rute yang berbeda tersebut yaitu (1) rute terpendek dari Terminal Banyuputih ke Tubing Pandansari di mana menggunakan algoritma Dijkstra diperoleh rute $v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_6 \rightarrow v_7 \rightarrow v_{11}$ dengan waktu perjalanan 41,38 menit dan panjang jalan 40 kilometer, sedangkan menggunakan algoritma Floyd-Warshall diperoleh rute $v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21} \rightarrow v_{13} \rightarrow v_{12} \rightarrow v_{11}$ dengan waktu perjalanan 51,26 menit dan panjang jalan 41 kilometer, (2) rute terpendek dari Terminal Banyuputih ke Bandar Ecopark di mana menggunakan algoritma Dijkstra diperoleh rute $v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21}$ dengan waktu perjalanan 28,27 menit dan panjang jalan 26,6 kilometer, sedangkan menggunakan algoritma Floyd-Warshall diperoleh rute $v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{27} \rightarrow v_{26} \rightarrow v_{21}$ dengan waktu perjalanan 36,78 menit dan panjang jalan 30 kilometer. Dapat disimpulkan bahwa algoritma Dijkstra lebih tepat untuk digunakan dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang.

DAFTAR ISI

Halaman

HALAMAN JUDUL i

PERNYATAAN KEASLIAN TULISAN ii

HALAMAN PENGESAHAN iii

MOTTO DAN PERSEMBAHAN iv

KATA PENGANTAR v

ABSTRAK vii

DAFTAR ISI viii

DAFTAR TABEL xi

DAFTAR GAMBAR xiii

DAFTAR LAMPIRAN xiv

BAB

1. PENDAHULUAN 1

    1.1. Latar Belakang 1

    1.2. Rumusan Masalah 8

    1.3. Batasan Masalah 8

    1.4. Tujuan 9

    1.5. Manfaat 10

    1.6. Sistematika Penulisan 10

2. TINJAUAN PUSTAKA 12

    2.1. Riset Operasi 12

    2.2. Analisis Jaringan 13

UNNES
UNIVERSITAS NEGERI SEMARANG

2.3. Pengertian Graf 14
2.4. Terminologi Graf 15
2.5. Jenis-jenis Graf 17
2.6. Keterhubungan 20
2.7. Representasi Graf dalam Matriks 24
2.8. Rute Terpendek (*Shortest Route*) 27
2.9. Algoritma Dijkstra 27
2.10. Algoritma Floyd-Warshall 34
2.11. Jalan 42
2.12. Volume Lalu Lintas 43
2.13. Kapasitas Jalan 44
2.14. Tingkat Pelayanan (V/C Ratio) 45
2.15. Kepadatan 47
2.16. Kemacetan 48
3. METODE PENELITIAN 51
3.1. Studi Pustaka 51
3.2. Pengumpulan Data 51
3.3. Pemecahan Masalah 52
3.4. Penarikan Kesimpulan 53
4. HASIL PENELITIAN DAN PEMBAHASAN 54
4.1. Hasil Penelitian 54
4.2. Pembahasan 81

UNNES
UNIVERSITAS NEGERI SEMARANG

5. PENUTUP 86

5.1. Simpulan 86

5.2. Saran 90

DAFTAR PUSTAKA 92

LAMPIRAN 94

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR TABEL

Tabel Halaman

2.1. Tabel hasil iterasi menggunakan Algoritma Dijkstra 32

2.2. Tabel hasil rute terpendek menggunakan Algoritma Floyd-Warshall ... 41

2.3. Tabel karakteristik tingkat pelayanan beserta batas lingkupnya 46

4.1. Tabel titik yang merepresentasikan stasiun/terminal di Batang, persimpangan jalan dan tempat-tempat wisata di Batang 56

4.2. Tabel sisi yang merepresentasikan jalan yang dilalui wisatawan dari stasiun/terminal di Batang menuju ke tempat wisata di Batang 56

4.3. Tabel bobot sisi yang merepresentasikan waktu perjalanan yang dibutuhkan dari suatu titik ke titik lain 58

4.4. Tabel penyelesaian dari penerapan Algoritma Dijkstra dalam menentukan rute terpendek dari Stasiun Batang Baru (v_6) ke semua tempat wisata di Batang $(v_1, v_2, v_3, v_{11}, v_{19}, v_{21}, v_{27}, v_{29}, v_{32})$ 71

4.5. Tabel penyelesaian dari penerapan Algoritma Dijkstra dalam menentukan rute terpendek dari Terminal Banyuputih (v_{17}) ke semua tempat wisata di Batang $(v_1, v_2, v_3, v_{11}, v_{19}, v_{21}, v_{27}, v_{29}, v_{32})$ 72

4.6. Tabel penyelesaian dari penerapan Algoritma Dijkstra dalam menentukan rute terpendek dari Terminal Limpung (v_{24}) ke semua tempat wisata di Batang $(v_1, v_2, v_3, v_{11}, v_{19}, v_{21}, v_{27}, v_{29}, v_{32})$ 73

4.7. Tabel penyelesaian dari penerapan Algoritma Floyd-Warshall dalam menentukan rute terpendek dari Stasiun Batang Baru (v_6) ke semua tempat wisata di Batang $(v_1, v_2, v_3, v_{11}, v_{19}, v_{21}, v_{27}, v_{29}, v_{32})$ 78

UNNES
UNIVERSITAS NEGERI SEMARANG

4.8. Tabel penyelesaian dari penerapan Algoritma Floyd-Warshall dalam menentukan rute terpendek dari Terminal Banyuputih (v_{17}) ke semua tempat wisata di Batang ($v_1, v_2, v_3, v_{11}, v_{19}, v_{21}, v_{27}, v_{29}, v_{32}$) 79
4.9. Tabel penyelesaian dari penerapan Algoritma Floyd-Warshall dalam menentukan rute terpendek dari Terminal Limpung (v_{24}) ke semua tempat wisata di Batang ($v_1, v_2, v_3, v_{11}, v_{19}, v_{21}, v_{27}, v_{29}, v_{32}$) 80
4.10. Tabel perbandingan hasil rute terpendek yang diperoleh dari penerapan Algoritma Dijkstra dan Floyd-Warshall 81
4.11. Tabel Perbedaan Algoritma Dijkstra dan Floyd-Warshall 83
4.12. Tabel rute menuju suatu tempat wisata yang merupakan bagian dari rute menuju tempat wisata lain 84
4.13. Tabel Paket Wisata di Batang 85
5.1. Tabel hasil penerapan Algoritma Dijkstra dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang 87
5.2. Tabel hasil penerapan Algoritma Floyd-warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang 88
5.3. Tabel rute terpendek yang direkomendasikan dari stasiun/terminal di Batang menuju ke tempat wisata di Batang..... 89

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR GAMBAR

Gambar Halaman

2.1. Terminologi Graf 16

2.2. Jenis Graf berdasarkan ada tidaknya sisi-rangkap atau gelung 18

2.3. Jenis Graf berdasarkan orientasi arah pada sisi graf 19

2.4. Jalan, Jejak, Lintasan, Sirkit dan Sikel 22

2.5. Graf Kosong dan Graf Berlabel (Graf-bobot) 23

2.6. Graf G 24

2.7. Graf Berarah Berlabel G 29

2.8. Graf Berarah Berlabel G 35

2.9. Grafik Kapasitas Jalan 45

4.1. Graf Berbobot Rute Tempat Wisata di Batang 60

4.2. Matriks Berhubungan-langsung Graf Berbobot Rute Tempat Wisata di Batang 61

5.1. Graf Berbobot Rute Tempat Wisata di Batang 86

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR LAMPIRAN

Lampiran Halaman

1. Peta Wisata Kabupaten Batang 94
2. Tempat Wisata Unggulan di Kabupaten Batang 95
3. Peta Jaringan Jalan di Kabupaten Batang 97
4. Periode dan Volume Jam Sibuk Ruas Jalan di Kabupaten Batang 98
5. Kapasitas Ruas Jalan di Kabupaten Batang 99
6. V/C Ratio Ruas Jalan di Kabupaten Batang 102
7. Kepadatan Ruas Jalan di Kabupaten Batang 105
8. Perhitungan untuk Menentukan Bobot Sisi (Waktu Perjalanan) Graf Rute Tempat Wisata di Batang 108
9. Graf Berbobot Rute Tempat Wisata di Batang 111
10. Matriks Berhubungan-langsung W untuk Menyatakan Graf Berbobot Rute Tempat Wisata di Batang 112
11. Hasil Iterasi Penerapan Algoritma Dijkstra dalam Menentukan Lintasan Terpendek dari Stasiun Batang Baru ke Semua Tempat Wisata di Batang 113
12. Penyelesaian dari Penerapan Algoritma Dijkstra dalam Menentukan Lintasan Terpendek dari Stasiun Batang Baru ke Semua Tempat Wisata di Batang 114
13. Hasil Iterasi Penerapan Algoritma Dijkstra dalam Menentukan Lintasan Terpendek dari Terminal Banyuputih ke Semua Tempat Wisata di Batang 115
14. Penyelesaian dari Penerapan Algoritma Dijkstra dalam Menentukan Lintasan Terpendek dari Terminal Banyuputih ke Semua Tempat Wisata di Batang 116
15. Hasil Iterasi Penerapan Algoritma Dijkstra dalam Menentukan Lintasan Terpendek dari Terminal Limpung ke Semua Tempat Wisata di Batang 117

16. Penyelesaian dari Penerapan Algoritma Dijkstra dalam Menentukan Lintasan Terpendek dari Terminal Limpung ke Semua Tempat Wisata di Batang 118
17. Nilai $W[i,j]$ yang Perlu Dihitung Yaitu Setiap Sel Matriks W yang Tidak Diarsir 119
18. Hasil Iterasi untuk k = (1, 2, ..., 32) 120
19. Hasil Iterasi Penerapan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek dari stasiun/terminal di Batang menuju ke Semua Tempat Wisata di Batang 152
20. Penyelesaian dari Penerapan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek dari Stasiun Batang Baru ke Semua Tempat Wisata di Batang 153
21. Penyelesaian dari Penerapan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek dari Terminal Banyuputih ke Semua Tempat Wisata di Batang 154
22. Penyelesaian dari Penerapan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek dari Terminal Limpung ke Semua Tempat Wisata di Batang 155

UNNES
UNIVERSITAS NEGERI SEMARANG

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Matematika adalah cabang ilmu pengetahuan yang sangat penting dan sangat berperan dalam perkembangan dunia. Matematika dibandingkan dengan disiplin-disiplin ilmu yang lain mempunyai karakteristik tersendiri. Pentingnya matematika tidak lepas dari perannya dalam segala jenis dimensi kehidupan. Selain itu, matematika juga seringkali dibutuhkan untuk menunjang eksistensi ilmu-ilmu lain seperti fisika, kimia, astronomi, biologi, ekonomi, dan lain sebagainya. Tidak heran mengapa matematika dijuluki "*mathematics is a queen, but also a servant*", matematika sebagai ratu ilmu, tetapi juga sekaligus pelayan ilmu-ilmu lain. Matematika dikatakan sebagai ratu ilmu karena matematika dapat tumbuh dan berkembang untuk dirinya sendiri sebagai suatu ilmu tanpa adanya bantuan dari ilmu lain (Bell, 1952).

Matematika sebagai pengetahuan dasar telah memberikan kemajuan yang begitu banyak dalam berbagai bidang. Teori graf merupakan salah satu cabang matematika yang turut memberikan andil dalam kemajuan tersebut. Tokoh-tokoh yang berperan penting dalam teori graf diantaranya, Euler yang berisi tentang upaya pemecahan masalah jembatan Konigsberg yang sangat terkenal di Eropa. Tahun 1847, G.R Kirchoff (1824-1887) berhasil mengembangkan teori pohon (*theory of trees*) yang digunakan dalam persoalan jaringan listrik. Sepuluh tahun

UNNES
UNIVERSITAS NEGERI SEMARANG

kemudian A. Cayley (1821-1895) menemukan konsep pohon untuk menjelaskan permasalahan kimia yaitu hidrokarbon. Pada tahun 1859, Sir W.R Hamilton (1805-1865) berhasil menemukan suatu permainan yang kemudian dijualnya ke pabrik mainan di Dublin. Permainan tersebut dari kayu berbentuk *dodecahedron* beraturan yakni berupa sebuah pentagon beraturan dan tiap pojoknya dibentuk oleh tiga sisi berbeda. Tiap pojok dari *dodecahedron* tersebut dipasangkan dengan kota terkenal seperti London, New York, Paris, dan lain-lain. Masalah dalam permainan ini adalah diminta untuk mencari suatu rute melalui sisi-sisi dari *dodecahedron* sehingga tiap kota dari 20 kota yang ada dapat dilalui tepat satu kali.

Teori graf mengalami perkembangan yang sangat pesat. Faktor yang mempercepat perkembangan ini adalah kemajuan teknologi komputer. Penggunaannya yaitu di dalam masalah optimasi skala besar yang dapat dimodelkan dalam bentuk graf dan dipecahkan melalui algoritma yang diberikan oleh teori graf (Kreyszig, 1993).

Hasil penelitian Mardlootillah dkk (2015) menunjukkan bahwa simulasi algoritma Dijkstra menggunakan *Visual Basic* dapat diimplementasikan pada permasalahan sehari-hari dalam menangani masalah rute terpendek.

Salah satu persoalan yang sering ditemui dalam kehidupan sehari-hari adalah pencarian rute terpendek (*shortest route*). Persoalan ini bisa dimodelkan ke dalam suatu graf berbobot dengan nilai pada masing-masing sisi yang mewakili persoalan yang akan dipecahkan. Kata “terpendek” pada kata “rute terpendek” ini tidak hanya diartikan sebagai jarak secara fisik, namun hal itu tergantung dari tipe

permasalahan yang akan dipecahkan. Bisa jadi kata tersebut memiliki makna tingkat aksesbilitas suatu titik dalam graf dari titik lain (Novandi, 2007).

Menurut Faro & Giordano (2016), penentuan rute terpendek dari kota A ke kota B tidak hanya ditentukan oleh jarak saja. Akan tetapi mempertimbangkan kondisi lalu lintas yang sebenarnya sehingga tidak hanya tentang rute terpendek tetapi juga tentang alternatif untuk menghindari kemacetan lalu lintas.

Masalah pencarian rute terpendek ini sangat rumit, apalagi jika yang dicari tersebut dalam skala besar, sehingga diperlukan pemecahan masalah dengan suatu algoritma. Beberapa algoritma yang dapat digunakan untuk memecahkan permasalahan rute terpendek yaitu algoritma Dijkstra, Bellman-Ford, Floyd-Warshall, Breadth First Search, A-Star, dan sebagainya. Pada permasalahan mencari rute terpendek ini, penulis menggunakan algoritma Dijkstra dan Floyd-Warshall.

Algoritma Dijkstra ditemukan oleh seorang ilmuan komputer berkebangsaan Belanda bernama Edsger Dijkstra. Cara kerja algoritma ini menggunakan prinsip *greedy*, di mana pada setiap langkah dipilih sisi dengan bobot minimum yang menghubungkan sebuah titik yang sudah terpilih dengan titik lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal dan tempat tujuan. Hasil akhir dari algoritma ini adalah panjang rute terpendek dari tempat asal ke tempat tujuan beserta rutenya (Munir, 2010).

Algoritma Floyd-Warshall bekerja berdasarkan formulasi *dynamic programing*, di mana setiap langkahnya akan menghitung bobot terkecil dari semua rute yang menghubungkan sebuah pasang titik, dan melakukannnya

sekaligus untuk semua pasangan titik. Dengan kata lain pada saat perhitungan rute terpendek, terlebih dahulu menghitung semua kemungkinan rute yang akan dilalui. Kemudian baru mencari rute terpendek dengan cara membandingkan tiap pasang rute dan dicari rute yang lebih pendek (Pandey, 2008).

Algoritma Dijkstra adalah algoritma yang sangat terkenal untuk memecahkan masalah rute terpendek. Ketika diterapkan pada situasi nyata, meskipun rute terpendek dapat dihitung dengan algoritma Dijkstra, hasil yang diperoleh tidak selalu menjadi salah satu yang dipilih. Dalam situasi lalu-lintas misalnya, pengemudi mungkin tidak tahu tepat rute terpendek untuk diikuti. Bahkan yang lebih menarik adalah kenyataan bahwa meskipun pengemudi tahu rute terpendek, dia mungkin lebih suka memilih rute yang berbeda (Galan-Garcia *et al.*, 2014).

Hasil penelitian Peyer *et al*. (2009) menunjukkan bahwa algoritma Dijkstra dapat digeneralisasikan untuk menemukan rute terpendek pada graf berarah dengan panjang sisi non-negatif. Seperti pelabelan suatu titik, label subgraf dari partisi grafik yang diberikan. Waktu berjalan dapat dicapai lebih baik jika jumlah subgraf yang terlibat lebih kecil dibandingkan dengan urutan graf asli dan jika masalah rute terpendek dibatasi untuk subgraf maka hasilnya akan lebih mudah dihitung.

Algoritma Floyd-Warshall adalah algoritma yang banyak digunakan untuk menghitung rute terpendek antara semua titik dalam graf berarah yang berbobot. Hali ini juga dapat digunakan untuk mendeteksi keberadaan siklus negatif (Hougardy, 2010).

UNNES
UNIVERSITAS NEGERI SEMARANG

Menurut Aini & Salehipour (2012), pada jaringan dengan siklus di mana setidaknya ada satu siklus, algoritma Floyd-Warshall adalah salah satu algoritma yang paling banyak digunakan. Algoritma ini digunakan untuk menentukan rute biaya antara setiap pasang titik.

Kabupaten Batang merupakan salah satu kabupaten di Jawa Tengah yang berada pada rute utama yang menghubungkan Jakarta-Surabaya. Arus transportasi dan mobilitas yang tinggi di rute pantura memberikan kemudahan untuk menuju Kabupaten Batang. Kondisi wilayah Kabupaten Batang merupakan kombinasi antara daerah pantai, dataran rendah, dan pegunungan yang kaya akan sumber daya alam, hutan, dan laut. Dengan kondisi ini Kabupaten Batang mempunyai potensi yang sangat besar dan sangat strategis untuk dikembangkan sebagai daerah wisata. Kabupaten Batang mempunyai banyak tempat dan jenis wisata. Dari wisata alam seperti Pantai Ujungnegoro, Pantai Sigandu, dan tempat wisata alam lain, wisata agro seperti Agrowisata Pagilaran dan Wisata Agro Selopajang Timur, wisata rekreasi dan pertunjukan seperti Batang Dolphins Center, Pemandian Air Hangat Sangubanyu, dan tempat wisata rekreasi lain.

Pemerintah Daerah Kabupaten Batang memberikan informasi tentang deskripsi, tiket masuk, peta lokasi, data kunjungan, dan foto-foto tempat wisata pada situs web yang dibuat oleh Dinas Kebudayaan dan Pariwisata Kabupaten Batang. Pemerintah daerah juga telah berusaha memberikan informasi sebagai penuntun wisatawan untuk menuju tempat wisata tertentu. Akan tetapi tidak bisa terjangkau secara detail karena sifatnya hanya sebagai petunjuk jalan, belum bisa

mengarahkan ke suatu lokasi tertentu. Minimnya fasilitas petunjuk arah ini menjadikan para wisatawan kesulitan untuk mencari tempat wisata.

Tempat wisata di Batang tersebar di berbagai penjuru Batang. Untuk menuju ke tempat wisata tersebut, ada beberapa rute yang bisa ditempuh. Wisatawan pasti menginginkan rute yang paling efisien untuk menuju tempat wisata tujuan sehingga dapat menghemat waktu dan biaya. Akan tetapi banyak wisatawan yang tidak mengetahui rute-rute untuk mengakses tempat wisata di Batang.

Pencarian rute yang paling efisien untuk menuju tempat wisata ini menuntut wisatawan untuk membuat keputusan rute mana yang diambil dari beberapa pilihan rute yang ada. Dalam menyelesaikan permasalahan yang berkaitan dengan pengambilan keputusan, dalam ilmu matematika diperlukan suatu teknik pemecahan masalah yaitu riset operasi. Riset operasi mencangkup banyak permasalahan di antaranya yaitu pemrograman linear, dualitas dan sensitivitas, analisis jaringan, perencanaan dan pengendalian proyek, pemrograman dinamis, teori permainan, pemrograman bilangan bulat, pemrograman tak linear, teori probabilitas, teori antrian, dan lain sebagainya.

UNNES
UNIVERSITAS NEGERI SEMARANG

Penentuan rute yang paling efisien untuk menuju tempat wisata ini termasuk ke dalam permasalahan analisis jaringan. Analisis jaringan ini terbagi menjadi tiga macam permasalahan yaitu permasalahan rute terpendek (*shortest route*), pohon rentang minimal (*minimum spanning tree*), dan aliran maksimum (*maximum flow*). Karena yang dicari adalah rute yang paling efisien maka permasalahan yang digunakan yaitu permasalahan rute terpendek.

Permasalahan rute terpendek yaitu menemukan rute terpendek antara dua atau beberapa titik lebih yang berhubungan. Permasalahan mencari rute terpendek merupakan salah satu permasalahan optimasi. Permasalahan ini biasanya direpresentasikan dalam bentuk graf. Graf yang digunakan dalam pencarian rute terpendek adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf ini merepresentasikan permasalahan yang akan dipecahkan.

Peralatan yang paling sering digunakan dalam mencari rute terpendek adalah peta atau *google map*. Peta atau *google map* dapat mengestimasikan rute yang optimal dengan menghitung jarak dan waktu tempuh normal. Karena merupakan estimasi, maka dengan cara tersebut belum tentu rute yang dipilih merupakan rute yang optimal karena terdapat faktor lain seperti tingkat kemacetan atau kondisi jalan. Terjadinya kemacetan di berbagai tempat menyebabkan waktu tempuh semakin lama. Sehingga untuk menentukan rute yang optimal selain melihat jarak dan waktu tempuh normal, perlu diperhatikan pula faktor lain seperti tingkat kemacetan atau kondisi jalan.

Berdasarkan latar belakang masalah tersebut maka penulis melakukan penelitian dengan judul "**Penerapan Algoritma Dijkstra dan Floyd-Warshall untuk Menentukan Rute Terpendek Tempat Wisata di Batang**".

UNNES
UNIVERSITAS NEGERI SEMARANG

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang diteliti adalah sebagai berikut.

1) Bagaimana bentuk graf dari rute tempat wisata di Batang?
2) Bagaimana penyelesaian dari penerapan algoritma Dijkstra dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang?
3) Bagaimana penyelesaian dari penerapan algoritma Floyd-Warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang?
4) Bagaimana rute terpendek yang direkomendasikan?

1.3. Batasan Masalah

Pada penelitian ini, penulis memberikan batasan-batasan permasalahan yang dibahas sebagai berikut.

1) Tempat wisata yang digunakan hanya tempat wisata unggulan yang ada di Batang.
2) Terminal yang digunakan yaitu Terminal Limpung dan Terminal Banyuputih serta stasiun yang digunakan yaitu Stasiun Batang Baru.
3) Ruas jalan yang digunakan yaitu ruas jalan arteri, ruas jalan kolektor, dan ruas jalan lokal.
4) Tempat wisata yang saat direpresentasikan dalam bentuk graf termasuk titik terasing, maka tempat wisata tersebut tidak digunakan.

UNNES
UNIVERSITAS NEGERI SEMARANG

5) Volume pada jam sibuk ruas jalan tertentu diabaikan.
6) Titik-titik pada graf merepresentasikan stasiun/terminal di Batang, persimpangan jalan, dan tempat-tempat wisata di Batang.
7) Sisi-sisi pada graf merepresentasikan jalan dari stasiun/terminal di Batang menuju ke tempat-tempat wisata di Batang.
8) Bobot sisi pada graf merepresentasikan waktu perjalanan.
9) Rute tempat wisata di Batang adalah rute yang dilalui wisatawan dari stasiun/terminal di Batang menuju ke tempat-tempat wisata di Batang tanpa perulangan.

1.4. Tujuan

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah sebagai berikut.

1) Membuat bentuk graf dari rute tempat wisata di Batang.
2) Menemukan penyelesaian dari penerapan algoritma Dijkstra dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang.
3) Menemukan penyelesaian dari penerapan algoritma Floyd-Warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang.
4) Menentukan rute terpendek yang direkomendasikan.

UNNES
UNIVERSITAS NEGERI SEMARANG

1.5. Manfaat

Manfaat yang diharapkan dalam penelitian ini adalah sebagai berikut.

1.5.1. Secara Teoritis

Sebagai sumber ilmu pengetahuan yang dijadikan bahan acuan untuk perluasan wawasan dalam bidang matematika terapan tentang masalah optimasi khususnya masalah rute terpendek.

1.5.2. Secara Praktis

Sebagai bahan informasi tentang rute terpendek menuju tempat wisata di Batang yang dapat digunakan untuk melengkapi situs web yang dibuat oleh Dinas Kebudayaan dan Pariwisata Kabupaten Batang serta sebagai bahan acuan bagi para wisatawan yang menginginkan rute yang paling efisien untuk menuju tempat wisata di Batang.

1.6. Sistematika Penulisan

Secara garis besar sistematika penulisan skripsi ini dibagi menjadi tiga bagian yaitu: (1) bagian awal, (2) bagian isi, dan (3) bagian akhir. Untuk memberikan gambaran yang jelas dan memudahkan pembaca menelaah isi skripsi ini maka disusun sistematika sebagai berikut.

1.6.1. Bagian Awal

Bagian awal berisi halaman judul, halaman pernyataan keaslian, halaman pengesahan, halaman motto dan persembahan, prakata, abstrak, daftar isi, daftar tabel, daftar gambar, dan daftar lampiran.

1.6.2. Bagian Isi

Bagian isi terdiri dari lima bab sebagai berikut.

a. BAB 1: Pendahuluan

Pendahuluan menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

b. BAB 2: Tinjauan Pustaka

Tinjauan pustaka merupakan dasar-dasar teori yang digunakan sebagai acuan dalam pemecahan dari permasalahan meliputi riset operasi, analisis jaringan, pengertian graf, terminologi graf, jenis-jenis graf, keterhubungan, representasi graf dalam matriks, rute terpendek (*shortest route*), algoritma Dijkstra, algoritma Floyd-Warshall, jalan, volume lalu lintas, kapasitas jalan, tingkat pelayanan (V/C ratio), kepadatan, dan kemacetan.

c. BAB 3: Metode Penelitian

Metode penelitian memaparkan tentang prosedur dan langkah-langkah yang dilakukan dalam penelitian yang terdiri dari studi pustaka, pengumpulan data, cara pemecahan masalah, dan penarikan kesimpulan.

d. BAB 4: Hasil Penelitian dan Pembahasan

Hasil penelitian dan pembahasan menguraikan tentang hasil penelitian dan pembahasan dari data yang diperoleh.

e. BAB 5: Penutup

Penutup berisi simpulan dan saran.

1.6.3. Bagian Akhir

Bagian akhir terdiri dari daftar pustaka dan lampiran.

UNNES
UNIVERSITAS NEGERI SEMARANG

BAB 2

TINJAUAN PUSTAKA

2.1. Riset Operasi

Riset operasi hampir digunakan dalam seluruh kegiatan, baik di perguruan tinggi, konsultan, rumah sakit, perencanaan kota, maupun pada kegiatan-kegiatan bisnis. Sebagai suatu teknik pemecahan masalah, riset operasi dipandang sebagai suatu ilmu dan seni. Aspek ilmu terletak pada penggunaan teknik-teknik dan algoritma-algoritma matematik untuk memecahkan permasalahan yang dihadapi, sedangkan sebagai seni ialah karena keberhasilan dari solusi model matematis ini sangat bergantung pada kreativitas dan kemampuan seseorang sebagai penganalisis dalam pengambilan keputusan (Dimyati, 2004: 1).

2.1.1. Komponen-Komponen Utama Permasalahan Keputusan

Menurut Dimyati (2004: 2), munculnya permasalahan-permasalahan keputusan adalah karena seorang pengambil keputusan sering dihadapkan pada beberapa pilihan tindakan yang harus dilakukan. Dalam menyelesaikan permasalahan yang berkaitan dengan pengambilan keputusan ini harus diidentifikasi terlebih dahulu dua komponen utamanya sebagai berikut.

1. *Objective* (tujuan)

   Tujuan (*objective*) yaitu hasil akhir yang hendak dicapai dengan cara memilih suatu tindakan yang paling tepat untuk sistem yang dipelajari.

UNNES
UNIVERSITAS NEGERI SEMARANG

2. Variabel-variabel

   Untuk dapat menentukan tindakan-tindakan yang mungkin dilakukan, harus diidentifikasi variabel-variabel sistem yang dapat dikendalikan oleh pengambil keputusan, yang keberhasilannya dalam mengidentifikasi variabel-variabel ini pun bergantung pada bias dan pelatihan si pengambil keputusan.

2.1.2. Metodologi Riset Operasi

Menurut Dimyati (2004: 4), jika riset operasi akan digunakan untuk memecahkan suatu permasalahan di suatu organisasi, maka harus dilakukan lima langkah sebagai berikut.

1. Memformulasikan Permasalahan
2. Mengobservasi sistem
3. Memformulasikan model matematis dari permasalahan yang dihadapi
4. Mengevaluasi model dan menggunakannya untuk prediksi
5. Mengimplementasi hasil studi

Riset operasi mencangkup banyak permasalahan diantaranya yaitu pemrograman linear, dualitas dan sensitivitas, analisis jaringan, perencanaan dan pengendalian proyek, pemrograman dinamis, teori permainan, pemrograman bilangan bulat, pemrograman tak linear, teori probabilitas, teori antrian, dan lain sebagainya. Yang dibahas di sini adalah permasalahan analisis jaringan.

2.2. Analisis Jaringan

Menurut Dimyati (2004: 162), analisis jaringan terbagi menjadi tiga macam permasalahan yaitu permasalahan rute terpendek (*shortest route*), pohon

rentang minimal (*minimum spanning tree*), dan aliran maksimum (*maximum flow*). Yang dibahas di sini adalah permasalahan rute terpendek.

Permasalahan rute terpendek yaitu menemukan rute terpendek antara dua atau beberapa titik lebih yang berhubungan. Permasalahan mencari rute terpendek merupakan salah satu permasalahan optimasi. Permasalahan ini biasanya direpresentasikan dalam bentuk graf. Graf yang digunakan dalam pencarian rute terpendek adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf ini merepresentasikan permasalahan yang akan dipecahkan.

2.3. Pengertian Graf

Sebuah graf G berisikan dua himpunan yaitu himpunan berhingga tak kosong V(G) dari objek-objek yang disebut titik dan himpunan berhingga (mungkin kosong) E(G) yang elemen-elemennya disebut sisi sedemikian hingga setiap elemen e dalam E(G) merupakan pasangan tak berurutan dari titik-titik di V(G). Himpunan V(G) disebut himpunan titik G dan himpunan E(G) disebut himpunan sisi G (Budayasa, 2007: 1).

UNNES
UNIVERSITAS NEGERI SEMARANG

Misalkan u dan v adalah dua titik di G dan e = (u,v) (sering ditulis e = uv) adalah sebuah sisi G. Titik u dan titik v berhubungan langsung (*adjacent*) di G, sisi e menghubungkan (*joining*) titik u dan titik v di G, u dan v titik-titik akhir sisi e, sisi e terkait (*incident*) dengan titik u dan juga titik v (Budayasa, 2007: 2).

2.4. Terminologi Graf

Ada beberapa terminologi dari teori graf yang digunakan untuk menjelaskan apa yang dilihat ketika melihat suatu graf.

2.4.1. Titik (*Vertex*)

Titik pada suatu graf disimbolkan dengan v dan dipresentasikan dengan sebuah noktah "." (Budayasa, 2007: 2).

2.4.2. Sisi (*Edge*)

Sisi yang menghubungkan dua titik pada suatu graf disimbolkan dengan e dan dipresentasikan dengan sebuah kurva sederhana (ruas garis) dengan titik-titik akhir di kedua titik tersebut (Budayasa, 2007: 2).

2.4.3. Sisi-rangkap/ Sisi-ganda (*Multiple-edges*)

Jika terdapat lebih dari satu sisi yang menghubungkan dua titik u dan v pada suatu graf, maka sisi-sisi tersebut disebut sisi-rangkap/ sisi-ganda (Budayasa, 2007: 3).

2.4.4. Gelung (*Loop*)

Sebuah sisi graf yang menghubungkan sebuah titik dengan dirinya sendiri disebut gelung (loop) (Budayasa, 2007: 3).

UNNES
UNIVERSITAS NEGERI SEMARANG

2.4.5. Derajat Titik

Misalkan G sebuah graf dan v sebuah titik G, derajat titik v dilambangkan dengan $d_G(v)$ atau d(v) adalah banyaknya sisi G yang terkait dengan titik v (dengan catatan setiap gelung dihitung dua kali) (Budayasa, 2007: 10).

Derajat minimum G dilambangkan dengan $\delta(G)$, didefinisikan sebagai berikut.

$$\delta(G) = minimum\,\{d(v)|\ v \epsilon\ V(G)\} \quad \text{..... (2.1)}$$

Sedangkan derajat maksimum G dilambangkan dengan $\Delta(G)$, didefinisikan sebagai berikut.

$$\Delta(G) = maksimum\,\{d(v)|\ v \epsilon\ V(G)\} \quad \text{..... (2.2)}$$

2.4.6. Ketetanggaan

Misalkan u dan v adalah dua titik di G, titik u dan v dikatakan bertetanggaan bila keduanya terhubung langsung di G.

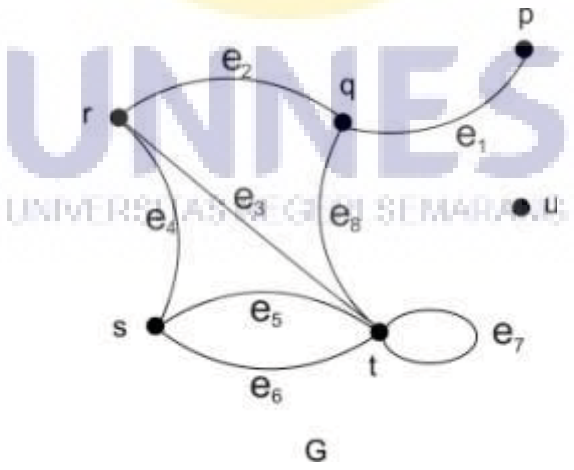
2.4.7. Bersisian

Misalkan u dan v adalah dua titik di G dan e = (u,v), maka dapat dikatakan e bersisian dengan titik u atau e bersisian dengan titik v.

2.4.8. Titik Terpencil

Titik terpencil adalah titik yang tidak mempunyai sisi yang bersisian dengannya.

Sebagai contoh, disajikan graf G berikut.



Gambar 2.1 Terminologi Graf

Keterangan:

Graf G dengan 6 titik dan 8 sisi;

V(G) = { p, q, r, s, t, u};

E(G) = { e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8};

e_1 = pq, e_2 = pr, e_3 = rt, e_4 = rs, e_5 = st, e_6 = st, e_7 = tt dan e_8 = tq;

Sisi-sisi e_5, e_6 disebut sisi-rangkap/ sisi-ganda;

Sisi e_7 disebut gelung;

d(p) = 1, d(q) = d(r) = d(s) = 3, d(t) = 6 dan d(u) = 0;

titik u disebut titik terpencil;

titik q bertetangga dengan titik p dan titik r, dan seterusnya;

sisi e_1 bersisian dengan titik p dan titik q, dan seterusnya.

2.5. Jenis-jenis Graf

Graf dapat dikelompokkan menjadi beberapa kategori (jenis) bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi-rangkap atau gelung dan berdasarkan orientasi arah pada sisi graf.

UNNES
UNIVERSITAS NEGERI SEMARANG

2.5.1. Jenis Graf Berdasarkan Ada Tidaknya Sisi-rangkap atau Gelung

Berdasarkan ada tidaknya sisi ganda atau gelung pada suatu graf, graf dibedakan menjadi dua jenis.

1) Graf Sederhana (*Simple Graph*)

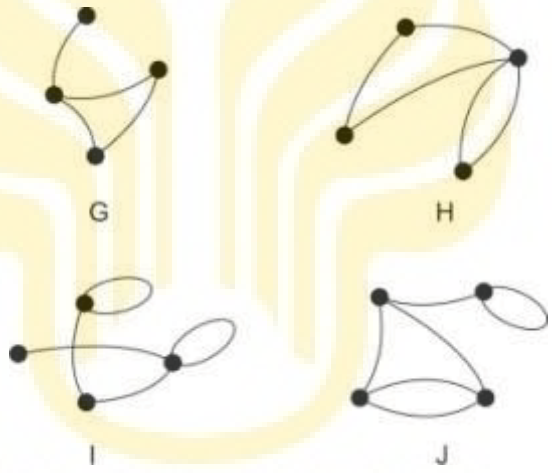
Graf yang tidak mempunyai sisi-rangkap dan tidak memiliki gelung disebut graf sederhana (Budayasa, 2007: 3).

2) Graf Tak Sederhana (*Unsimple Graph*)

Graf yang mengandung sisi rangkap atau gelung dinamakan graf tidak sederhana. Ada dua macam graf tak sederhana, yaitu graf rangkap (*multi graph*) dan graf semu (*pseudo graph*).

Graf rangkap (*multi graph*) adalah graf yang memiliki sisi-rangkap tetapi tidak memiliki gelung (Budayasa, 2007: 3). Graf semu (*pseudo graph*) adalah graf yang mengandung gelung (*loop*).

Sebagai contoh, disajikan graf G, H, I, dan J berikut.



Gambar 2.2 Jenis Graf berdasarkan ada tidaknya sisi-rangkap atau gelung

Keterangan:

G graf sederhana

H graf rangkap

I graf tak sederhana (graf semu tanpa sisi-rangkap)

J graf tak sederhana (graf semu dengan sisi-rangkap)

UNNES
UNIVERSITAS NEGERI SEMARANG

2.5.2. Jenis Graf Berdasarkan Orientasi Arah pada Sisi graf

Berdasarkan orientasi arah pada sisi graf, graf dibedakan menjadi dua jenis.

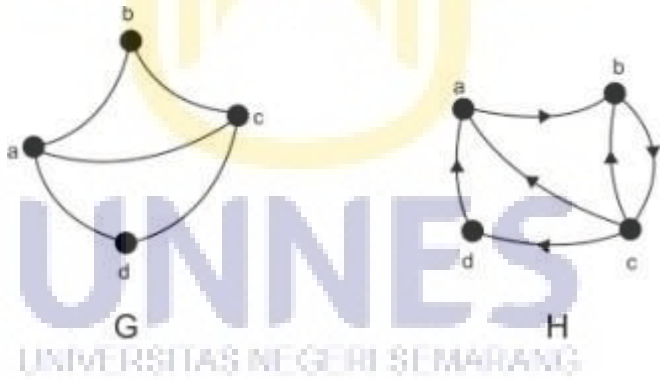
1) Graf Tak Berarah (*Undirected Graph*)

Graf tak berarah (*undirected graph*) adalah graf yang sisinya tidak mempunyai orientasi arah. Misalkan pada suatu graf G, sisi e = (u,v) merupakan sisi yang sama dengan e = (v,u).

2) Graf Berarah (*Directed Graph*)

Graf berarah (*directed graph*) adalah graf yang setiap sisinya diberikan orientasi arah. Misalkan pada suatu graf G, sisi e = (u,v) tidak sama dengan sisi e = (v,u).

Sebagai contoh, disajikan graf G dan H berikut.



Gambar 2.3 Jenis Graf berdasarkan orientasi arah pada sisi graf

Keterangan:

G graf tak berarah, dimana sisi (A,B) = (B,A)

H graf berarah, dimana sisi (A,B) ≠ (B,A)

2.6. Keterhubungan

Ada beberapa keterhubungan dari teori graf yang digunakan untuk menjelaskan suatu barisan yang suku-sukunya bergantian titik dan sisi pada suatu graf.

2.6.1. Jalan (*Walk*)

Misalkan G adalah sebuah graf. Sebuah jalan (*walk*) di G adalah sebuah barisan berhingga (tak kosong) $W = (v_0, e_1, v_1, e_2, v_2, \cdots, e_k, v_k)$ yang suku-sukunya bergantian titik dan sisi, sedemikian hingga v_{i-1} dan v_i adalah titik-titik akhir sisi e_i, untuk $1 \leq i \leq k$. *W* adalah sebuah jalan dari titik v_0 ke titik v_k atau jalan $-(v_0, v_k)$. Titik v_0 dan v_k berturut-turut disebut titik awal dan titik akhir *W*. Sedangkan titik-titik $v_1, v_2, \cdots, v_{k-1}$ disebut titik-titik internal *W*, dan *k* disebut panjang jalan *W*. Panjang jalan *W* adalah banyaknya sisi dalam *W*. Sebuah jalan *W* dengan panjang positif disebut tertutup jika titik awal dan titik akhir dari *W* identik (Budayasa, 2007: 6).

2.6.2. Jejak (*Trail*)

Misalkan G adalah sebuah graf. Jika terdapat sebuah jalan *W* dengan semua sisi $e_1, e_2, e_3, \cdots, e_k$ berbeda maka *W* disebut sebuah jejak (*trail*) (Budayasa, 2007: 6).

2.6.3. Rute (*Route*)

Misalkan G adalah sebuah graf. Jika terdapat sebuah jalan *W* dengan semua sisi $e_1, e_2, e_3, \cdots, e_k$ dan titik $v_0, v_1, v_2, \cdots, v_k$ berbeda maka *W* disebut sebuah rute (*route*) (Budayasa, 2007: 6).

2.6.4. Sirkit (Jejak Tutup)

Jejak yang berawal dan berakhir pada titik yang sama disebut sirkit. Sebuah sirkit dikatakan sirkit sederhana (*simple circuit*) jika sirkit tersebut tidak memuat atau melewati sisi yang sama dua kali (setiap sisi hanya dilewati satu kali). Sebuah sirkit dikatakan sirkit dasar (*elementary circuit*) jika sirkit tersebut tidak memuat atau melewati titik yang sama dua kali (setiap titik hanya dilewati satu kali, titik awal dan titik akhir boleh sama) atau bisa disebut **sikel (*cycle*).** Banyaknya sisi dalam suatu sikel disebut panjang dari sikel tersebut. Sikel dengan panjang k disebut sikel-k, disimbolkan dengan C_k. Sebuah sikel yang memuat semua titik sebuah graf disebut **sikel Hamilton**. Graf yang memuat sikel Hamilton disebut **graf Hamilton** (Budayasa, 2007: 6).

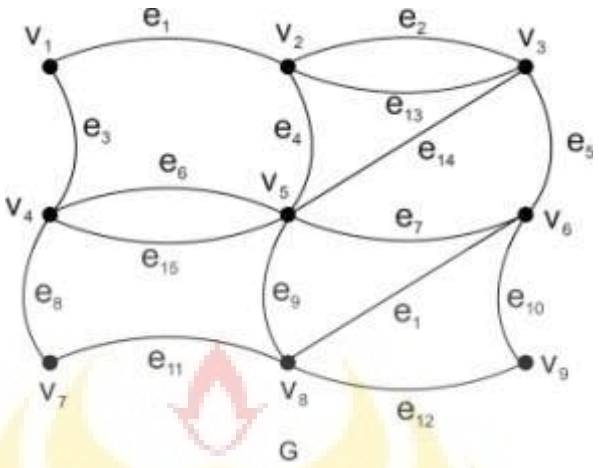
2.6.5. Sirkit Euler

Sirkit Euler adalah sebuah sirkit di graf G yang memuat semua sisi G. Sebuah graf yang memuat semua sirkit Euler disebut **graf Euler** (Budayasa, 2007: 6).

Misalkan G adalah suatu graf. Sirkit Euler G adalah sirkit dimana setiap titik dalam G muncul paling sedikit sekali dan setiap sisi dalam G muncul tepat satu kali (Siang, 2006: 213).

UNNES
UNIVERSITAS NEGERI SEMARANG

Sebagai contoh, disajikan graf G berikut.



Gambar 2.4 Jalan, Jejak, Rute, Sirkit, dan Sikel (Budayasa, 2007: 7)

Keterangan:

1) Barisan $(v_1, e_1, v_2, e_2, v_3, e_5, v_6, e_5, v_3)$ adalah sebuah jalan - (v_1, v_3) di graf G yang panjangnya 4. Karena dalam barisan ini sisi e_5 muncul lebih dari satu kali, jelas barisan ini bukan jejak.
2) Barisan $(v_1, e_3, v_4, e_6, v_5, e_9, v_8, e_{11}, v_7, e_8, v_4)$ adalah sebuah jejak buka di G dengan panjang 5. Karena titik v_4 muncul lebih dari satu kali, maka jejak tersebut bukan rute.
3) Barisan $(v_1, e_3, v_4, e_8, v_7, e_{11}, v_8, e_9, v_5)$ adalah sebuah rute di graf G dengan panjang 4.
4) Barisan $(v_1, e_1, v_2, e_4, v_5, e_9, v_8, e_{12}, v_9, e_{10}, v_6, e_7, v_5, e_6, v_4, e_3, v_1)$ adalah sebuah jejak tutup (sirkit) di G dengan panjang 8. Jejak tutup ini bukan sikel karena titik internal v_5 muncul lebih dari satu kali. Sirkit $(v_1, e_1, v_2, e_2, v_3, e_{13}, v_2, e_4, v_5, e_{14}, v_3, e_5, v_6, e_7, v_5,$

$e_9,\ v_8,\ e_{16},\ v_6,\ e_{10},\ v_9,\ e_{12},\ v_8,\ e_{11},\ v_7,\ e_8,\ v_4,\ e_6,\ v_5,\ e_{15},\ v_4,\ e_3,\ v_1)$ adalah sebuah sirkit Euler di G. jadi G adalah graf Euler.

5) Barisan $(v_1,\ e_3,\ v_4,\ e_6,\ v_5,\ e_4,\ v_2,\ e_1,\ v_1)$ adalah sebuah sikel di G dengan panjang 4. Sikel $(v_1,\ e_1,\ v_2,\ e_2,\ v_3,\ e_{14},\ v_5,\ e_7,\ v_6,\ e_{10},\ v_9,\ e_{12},\ v_8,\ e_{11},\ v_7,\ e_8,\ v_4,\ e_3,\ v_1)$ memuat semua titik G, jadi sikel tersebut merupakan sikel Hamilton. Dengan demikian graf G merupakan graf Hamilton.

2.6.6. Graf Kosong (Graf Nol)

Graf kosong (graf nol) yaitu graf yang tidak memiliki sisi. Graf nol dengan *n* titik dilambangkan (N_n) (Budayasa, 2007: 3).

2.6.7. Graf Berlabel (Graf-bobot)

Graf berlabel/ berbobot adalah sebuah graf yang setiap sisinya dikaitkan dengan suatu bilangan real. Bilangan yang dikaitkan ke suatu sisi G disebut bobot sisi tersebut. Bobot graf G dilambangkan *w*(G) adalah jumlah bobot semua sisi G.

Sebagai contoh, disajikan graf G dan H berikut.

Gambar 2.5 Graf Kosong dan Graf Berlabel (Graf-bobot)

Keterangan:

G graf kosong dengan 3 titik

H graf-bobot dengan w(H) = 12

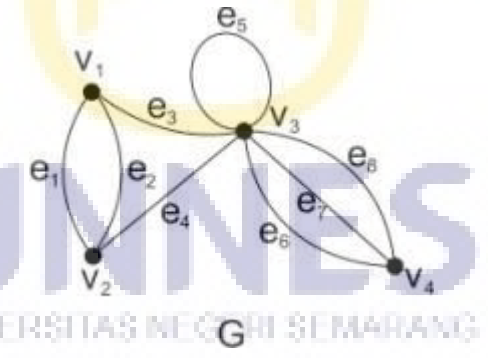
2.7. Representasi Graf dalam Matriks

Suatu graf dapat disajikan dalam bentuk matriks. Ada dua jenis matriks yang dapat digunakan yaitu matriks berhubungan-langsung (*adjacency matrix*) dan matriks keterkaitan (*incidence matrix*).

2.7.1. Matriks Berhubungan-Langsung (*Adjacency Matrix*)

Misalkan G sebuah graf dengan V(G) = $\{ v_1, v_2, \cdots, v_n \}$. Matriks berhubungan-langsung graf G adalah matriks bujur sangkar A(G) = (a_{ij}) berordo $n \times n$ yang baris-baris dan kolom-kolomnya dilabel dengan label titik-titik graf G sedemikian hingga elemen a_{ij} menyatakan banyaknya sisi G yang menghubungkan titik v_i dan v_j (Budayasa, 2007: 13).

Sebagai contoh, disajikan graf G berikut.



Gambar 2.6 Graf G

Matriks berhubungan-langsung graf G dapat ditulis sebagai berikut.

$$A(G) = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & V_3 & V_4 \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 1 & 3 \\ 0 & 0 & 3 & 0 \end{bmatrix} \end{array}$$

Dapat dilihat bahwa A(G) adalah matriks simetris dan unsur-unsurnya bilangan bulat non negatif. Jika G tidak memiliki gelung, maka setiap unsur A(G) yang terletak di diagonal utama adalah nol. Kalau G graf sederhana, maka unsur-unsur A(G) nol atau satu. Jika diberikan sebuah matriks simetris A berordo $n \times n$ dan unsur-unsurnya bilangan bilangan bulat non negatif, maka pasti ada graf G dengan n titik dan matriks berhubungan-langsung A(G) = A.

Dapat dilihat bahwa derajat titik graf G diperoleh dengan menjumlahkan semua unsur A yang terletak di baris yang bersesuaian dengan titik tersebut. Setelah unsur pada diagonal utama di baris itu dikalikan dua, dari matriks A(G) diperoleh:

d(v_1) = 2×0 + 2 + 1 + 0 = 3,

d(v_2) = 2 + 2×0 + 1 + 0 = 3,

d(v_3) = 1 + 1 + 2×1 + 3 = 7,

d(v_4) = 0 + 0 + 3 + 2×0 = 3.

Jadi, jika A = (a_{ij}) matriks berhubungan-langsung graf G maka:

$d(v_i) = a_{ij} + \sum_{j=1}^{n} a_{ij}$ ..... (2.3)

Jika matriks A(G) = A dikalikan dengan dirinya sendiri, maka:

$$A^2 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & V_3 & V_4 \end{array} \\ \begin{bmatrix} 5 & 1 & 3 & 3 \\ 1 & 5 & 3 & 3 \\ 3 & 3 & 12 & 3 \\ 3 & 3 & 3 & 9 \end{bmatrix} \end{array}$$

Dapat dilihat bahwa unsur A^2 yang terletak di baris ke-*i* dan kolom ke-*j* menyatakan banyaknya jalan -(v_i, v_j) di graf G dengan panjang 2. Misalkan unsur A^2 yang terletak di baris ke-1 dan kolom ke-4 adalah 3. Jadi ada 3 jalan -(v_1, v_4)

di graf G dengan panjang 2, yaitu $(v_1,\ e_3,\ v_3,\ e_6,\ v_4)$, $(v_1,\ e_3,\ v_3,\ e_7,\ v_4)$ dan $(v_1,\ e_3,\ v_3,\ e_8,\ v_4)$. Ada 5 jalan -$(v_1, v_1)$ di graf G dengan panjang 2, yaitu $(v_1,\ e_1,\ v_2,\ e_1,\ v_1), (v_1,\ e_1,\ v_2,\ e_2,\ v_1), (v_1,\ e_2,\ v_2,\ e_1,\ v_1), (v_1,\ e_2,\ v_2,\ e_2,\ v_1)$, dan $(v_1,\ e_3,\ v_3,\ e_3,\ v_1)$. Begitu juga terdapat 3 jalan -(v_3, v_2) di graf G dengan panjang 2, yaitu $(v_3,\ e_3,\ v_1,\ e_2,\ v_2), (v_3,\ e_5,\ v_3,\ e_4,\ v_2)$, dan $(v_3,\ e_3,\ v_1,\ e_1,\ v_2)$. Generalisasi dari observasi ini diberikan pada teorema berikut.

Teorema: Misalkan G adalah graf dengan V(G) = $\{\ v_1, v_2, \cdots, v_n\}$. Jika A adalah matriks berhubungan-langsung graf G, maka untuk suatu bilangan bulat positif *k*, unsur matriks A^k yang terletak di baris ke-*i* dan kolom ke-*j* menyatakan banyaknya jalan yang menghubungkan titik v_i dan v_j di graf G dengan panjang *k* (Budayasa, 2007: 15).

Teorema: Misalkan G sebuah graf dengan V(G) = $\{\ v_1,\ v_2,\ \cdots,\ v_n\}$. Misalkan A adalah matriks berhubungan-langsung graf G dan B=(b_{ij}) adalah matriks dengan B = A + A^2 + ⋯ + A^{n-1}. Graf G terhubung jika dan hanya jika $b_{ij} \neq 0$, untuk setiap *i* dan *j*, $i \neq j$ (Budayasa, 2007: 16).

2.7.2. Matriks Keterkaitan (*Incidence Matrix*)

Menurut Budayasa (2007: 16), jika graf G mempunyai *n* buah titik: v_1, v_2, ⋯, v_n dan *t* buah sisi: e_1, e_2, ⋯, e_t, maka matriks keterkaitan (*incidence matrix*) graf G adalah matriks M(G)= (m_{ij}) berordo $n \times t$ yang baris-barisnya dilabel dengan label titik-titik G dan kolom-kolomnya dilabel dengan sisi-sisi G sedemikian hingga

$$m_{ij} = \begin{cases} 0, & jika\ sisi\ \ e_j\ tidak\ terkait\ dengan\ titik\ \ v_i \\ 1, & jika\ sisi\ \ e_j\ terkait\ dengan\ titik\ \ v_i\ dan\ \ e_j\ bukan\ gelung \\ 2, & jika\ sisi\ \ e_j\ terkait\ dengan\ titik\ \ v_i\ dan\ \ e_j\ \ gelung \end{cases}$$

UNNES
UNIVERSITAS NEGERI SEMARANG

Sebagai contoh, disajikan graf G pada gambar 2.6. Matriks keterkaitan graf tersebut adalah sebagai berikut.

$$M(G) = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Dapat dilihat bahwa jumlah semua unsur M(G) yang terletak pada baris ke-*i* menyatakan derajat titik v_i di graf G, sedangkan jumlah semua unsur M(G) yang terletak pada setap kolom selalu 2.

2.8. Rute Terpendek (*Shortest Route*)

Panjang rute dalam sebuah graf-bobot adalah jumlah bobot semua sisi pada rute tersebut. Misalkan *u* dan *v* dua titik di graf G. rute –(*u,v*) di G dengan panjang minimum adalah rute terpendek antara *u* dan *v*. Sedangkan panjang rute dari *u* ke *v* dinotasikan dengan $d_G(u, v)$ atau d(*u,v*), didefinisikan sebagai panjang rute terpendek antara titik *u* dan titik *v* di G (Budayasa, 2007: 47).

Permasalahan mencari rute terpendek ini dapat dipecahkan dengan berbagai algoritma. Algoritma yang digunakan dalam penelitian ini adalah algoritma Dijkstra dan algoritma Floyd-Warshall.

2.9. Algoritma Dijkstra

Misalkan G adalah graf berlabel dengan titik-titik V(G) = { $v_1, v_2, \cdots, v_n$} dan rute terpendek yang dicari adalah dari v_1 ke v_n. Algoritma Dijkstra dimulai dari titik v_1. Dalam iterasinya, algoritma ini akan mencari satu titik yang jumlah

bobotnya terkecil. Titik-titik yang terpilih dipisahkan dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya (Siang, 2004: 276).

Misalkan:

V(G) = $\{ v_1, v_2, \cdots, v_n \}$

L = himpunan titik-titik ∈ V(G) yang sudah terpilih dalam alur rute terpendek

D(j) = jumlah bobot rute terkecil dari v_1 ke v_j

W(i,j) = bobot sisi dari titik v_i ke titik v_j

W*(1,j) = jumlah bobot rute terkecil dari v_1 ke v_j

Menurut Siang (2004), algoritma Dijkstra untuk mencari rute terpendek adalah sebagai berikut.

1) L = { }

   V = $\{ v_2, v_3, \cdots, v_n \}$

2) Untuk i = 2, ⋯, n, lakukan D (i) = W (1,i)

3) Selama $v_n \notin$ L lakukan:

   a) Pilih titik $v_k \in$ V – L dengan D(k) terkecil

      L = L ∪ $\{ v_k \}$

   b) Untuk setiap $v_j \in$ V – L lakukan:

      Jika D(j) > D(k) + W(k,j) maka ganti

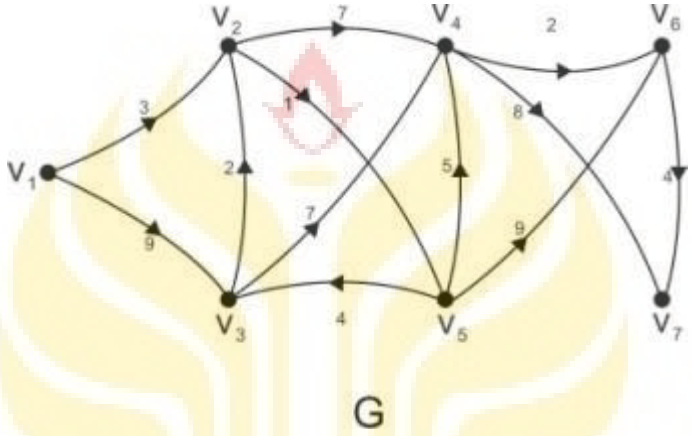
      D(j) dengan D(k) + W(k,j)

4) Untuk setiap $v_j \in$ V, W*(1,j) = D(j)

UNNES
UNIVERSITAS NEGERI SEMARANG

Menurut algoritma di atas, rute terpendek dari titik v_1 ke v_n adalah melalui titik-titik dalam L secara berurutan, dan jumlah bobot rute terkecilnya adalah D(n) (Siang, 2004: 277).

Sebagai contoh, akan dicari rute terpendek dari titik v_1 ke titik v_7 pada graf berarah berlabel G berikut.



Gambar 2.7 Graf Berarah Berlabel G

Matriks berhubungan-langsung W untuk menyatakan graf G pada Gambar 2.7 adalah sebagai berikut.

$$W = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{array} \\ \begin{bmatrix} \infty & 3 & 9 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 7 & 1 & \infty & \infty \\ \infty & 2 & \infty & 7 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 2 & 8 \\ \infty & \infty & 4 & 5 & \infty & 9 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix} \end{array}$$

Mula-mula L = { } dan V(G) = { $v_2, v_3, \cdots, v_7$}

D(2) = W(1,2) = 3

D(3) = W(1,3) = 9

D(4) = W(1,4) = ∞

D(5) = W(1,5) = ∞

D(6) = W(1,6) = ∞

D(7) = W(1,7) = ∞

V – L = { $v_2, v_3, \cdots, v_7$} – { } = { $v_2, v_3, \cdots, v_7$}

$v_n = v_7 \notin$ L, sehingga langkah 3(a) – 3(b) dilakukan

3(a) : D(*k*) terkecil adalah D(2), sehingga $v_k = v_2$

L = L ∪ { v_k} = { } ∪ { v_2} = { v_2}

3(b) : V – L = { $v_2, v_3, v_4, v_5, v_6, v_7$} - { v_2} = { v_3, v_4, v_5, v_6, v_7}

k = 2 (dari langkah 3(a))

Diselidiki tiap titik dalam V – L

Untuk *j* = 3

D(*j*) = D(3) = 9

D(*k*) + W(*k,j*) = D(2) + W(2,3) = 3 + ∞ = ∞

Karena D(3) ≯ D(2) + W(2,3) maka D(3) tetap yaitu 9.

Untuk *j* = 4

D(*j*) = D(4) = ∞

D(*k*) + W(*k,j*) = D(2) + W(2,4) = 3 + 7 = 10

Karena D(4) > D(2) + W(2,4) maka D(4) diubah menjadi 10

Ini berarti bahwa untuk mencapai titik v_4 dari v_1, rute melalui v_2 yaitu $v_1\ v_2\ v_4$ atau D(2) + W(2,4) mempunyai bobot lebih kecil.

UNNES
UNIVERSITAS NEGERI SEMARANG

Untuk $j = 5$

$D(j) = D(5) = \infty$

$D(k) + W(k,j) = D(2) + W(2,5) = 3 + 1 = 4$

Karena $D(5) > D(2) + W(2,5)$ maka D(5) diubah menjadi 4.

Untuk $j = 6$

$D(j) = D(6) = \infty$

$D(k) + W(k,j) = D(2) + W(2,6) = 3 + \infty = \infty$

Karena $D(6) \not> D(2) + W(2,6)$ maka D(6) tetap yaitu ∞.

Untuk $j = 7$

$D(j) = D(7) = \infty$

$D(k) + W(k,j) = D(2) + W(2,7) = 3 + \infty = \infty$

Karena $D(7) \not> D(2) + W(2,7)$ maka D(7) tetap yaitu ∞.

Langkah selanjutnya kembali ke langkah 3(a)

Karena $v_7 \notin L$

$V - L = \{ v_3, v_4, v_5, v_6, v_7 \}$

Di antara $D(k)$ ($k = 3,4,...,7$) hasil iterasi langkah 3(b), $D(k)$ yang terkecil adalah D(5), sehingga $v_k = v_5$

Maka sekarang $L = L \cup \{ v_5 \} = \{ v_2 \} \cup \{ v_5 \} = \{ v_2, v_5 \}$

3(b) : $V - L = \{ v_2, v_3, v_4, v_5, v_6, v_7 \} - \{ v_2, v_5 \} = \{ v_3, v_4, v_6, v_7 \}$

Langkah 3(b) untuk mengecek setiap titik dalam V – L diulang lagi.

Langkah 3(a) dan 3(b) diulang terus menerus hingga $v_7 \in L$.

Hasil itersi selengkapnya adalah sebagai berikut.

UNNES
UNIVERSITAS NEGERI SEMARANG

Tabel 2.1 Tabel hasil iterasi menggunakan Algoritma Dijkstra

Indeks *k* shg D(*k*) minimum	**L**	**V – L**	**D[2]**	**D[3]**	**D[4]**	**D[5]**	**D[6]**	**D[7]**
-	$\emptyset$	$\{v_2,v_3,v_4,v_5,v_6,v_7\}$	W(1,2)=3	W(1,3)=9	W(1,4)=∞	W(1,5)=∞	W(1,6)=∞	W(1,7)=∞
2	$\{v_2\}$	$\{v_3,v_4,v_5,v_6,v_7\}$	3 (tetap)	$Min(D(3),D(3)+W(2,3)) = min(9,3+\infty) = 9$	$Min(D(4),D(2)+W(2,4)) = min(\infty,3+7) = 10$	$Min(D(5),D(2)+W(2,5)) = min(\infty,3+1) = 4$	$Min(D(6),D(2)+W(2,6)) = min(\infty,3+\infty) = \infty$	$Min(D(7),D(2)+W(2,7)) = min(\infty,3+\infty) = \infty$
5	$\{v_2,v_5\}$	$\{v_3,v_4,v_6,v_7\}$	3 (tetap)	$Min(D(3),D(5)+W(5,3)) = min(9,4+4) = 8$	$Min(D(4),D(5)+W(5,4)) = min(10,4+5) = 9$	4 (tetap)	$Min(D(6),D(5)+W(5,6)) = min(\infty,4+9) = 13$	$Min(D(7),D(5)+W(5,7)) = min(\infty,4+\infty) = \infty$
3	$\{v_2,v_5,v_3\}$	$\{v_4,v_6,v_7\}$	3 (tetap)	8 (tetap)	$Min(D(4),D(3)+W(3,4)) = min(9,8+7) = 9$	4 (tetap)	$Min(D(6),D(3)+W(3,6)) = min(13,8+\infty) = 13$	$Min(D(7),D(3)+W(3,7)) = min(\infty,8+\infty) = \infty$
4	$\{v_2,v_5,v_3,v_4\}$	$\{v_6,v_7\}$	3 (tetap)	8 (tetap)	9 (tetap)	4 (tetap)	$Min(D(6),D(4)+W(4,6)) = min(13,9+2) = 11$	$Min(D(7),D(4)+W(4,7)) = min(\infty,9+8) = 17$
6	$\{v_2,v_5,v_3,v_4,v_6\}$	$\{v_7\}$	3 (tetap)	8 (tetap)	9 (tetap)	4 (tetap)	11 (tetap)	$Min(D(7),D(6)+W(6,7)) = min(17,11+4) = 15$
7	$\{v_2,v_5,v_3,v_4,v_6,v_7\}$	$\{\}$	3 (tetap)	8 (tetap)	9 (tetap)	4 (tetap)	11 (tetap)	15 (tetap)

Karena $v_n = v_7 \in$ L, maka iterasi dihentikan.

Rute terpendek dari v_1 ke v_7 adalah 15 dengan rute yang dibaca mundur sebagai berikut.

Pada iterasi dengan indeks *k* = 6, diperoleh penurunan panjang rute pada kolom D[7] dari 17 ke 15.

Ini berarti titik yang terpilih pada baris tersebut (v_6) adalah rute terpendek (rute $v_6 \rightarrow v_7$).

Berikutnya pada kolom D[6] diperoleh penurunan panjang rute dari 13 ke 11.

Ini berarti titik pada indeks *k* = 4 (v_4) adalah rute terpendek (rute $v_4 \rightarrow v_6 \rightarrow v_7$).

Pada kolom D[4] tidak terjadi penurunan panjang rute (dari 9 tetap 9).

Ini berarti titik pada indeks *k* = 3 (v_3) bukan rute terpendek.

Naik 1 baris diatasnya, terjadi penurunan panjang rute dari 10 ke 9 yaitu pada baris yang sesuai dengan indeks *k* = 5.

Jadi v_5 adalah rute terpendek (rute $v_5 \rightarrow v_4 \rightarrow v_6 \rightarrow v_7$).

Pada kolom D[5] terjadi penurunan panjang rute dari ∞ ke 4.

Baris tersebut sesuai dengan indeks *k* = 2.

Jadi v_2 adalah rute terpendek (rute $v_2 \rightarrow v_5 \rightarrow v_4 \rightarrow v_6 \rightarrow v_7$).

Jadi rute terpendek adalah $v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4 \rightarrow v_6 \rightarrow v_7$ dengan total panjang 15.

Hasil akhir dari algoritma Dijkstra adalah rute terpendek dari titik v_1 ke semua titik lain. Untuk mencari rute terpendek dari semua titik, algoritma Dijkstra dapat diulang-ulang sebanyak *n* kali yang dimulai dari semua titik.

UNNES
UNIVERSITAS NEGERI SEMARANG

2.10. Algoritma Floyd-Warshall

Masukan algoritma Floyd-Warshall adalah matriks berhubungan-langsung graf berlabel dan keluarannya adalah rute terpendek dari semua titik ke semua titik. Algoritma Floyd-Warshall memulai iterasi dari titik awalnya kemudian memperpanjang rute dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminim mungkin (Siang, 2004: 272).

Misalkan W_0 adalah matriks berhubungan-langsung graf berlabel mula-mula. W* adalah matriks berhubungan-langsung minimal dengan W_{ij}* = rute terpendek dari titik v_i ke titik v_j.

Menurut Siang (2004), algoritma Floyd-Warshall untuk mencari rute terpendek adalah sebagai berikut.

1) W = W_0

2) Untuk k = 1 hingga n, lakukan:

   untuk i = 1 hingga n, lakukan:

   untuk j = 1 hingga n, lakukan:

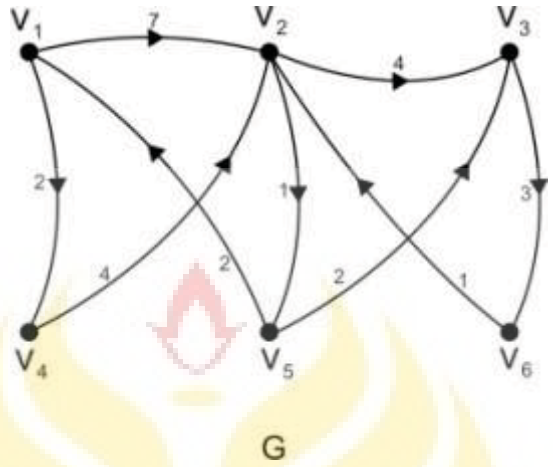
   jika $W[i,j] > W[i,k] + W[k,j]$ maka

   tukar $W[i,j]$ dengan $W[i,k] + W[k,j]$

3) W* = W

Dalam iterasinya untuk mencari rute terpendek, algoritma Floyd-Warshall membentuk n matriks, sesuai dengan iterasi-k (Siang, 2004: 273).

Sebagai contoh, akan dicari rute terpendek dari titik v_i ke titik v_j (i, j = 1, 2, ..., 6) pada graf berarah berlabel G berikut.



G

Gambar 2.8 Graf Berarah Berlabel G

Matriks berhubungan-langsung W untuk menyatakan graf G pada Gambar 2.8 adalah sebagai berikut.

$$W = W_0 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} \infty & 7 & \infty & 2 & \infty & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & \infty & \infty & \infty & \infty \\ 2 & \infty & 2 & \infty & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty \end{bmatrix} \end{array}$$

Iterasi untuk k = 1

Untuk setiap sel matriks W dicek apakah $W[i,j] > W[i,1] + W[1,j]$.

Jika ya, maka $W[i,j]$ diganti dengan $W[i,1] + W[1,j]$.

Sebagai contoh:

- $W[1,2] = 7$, sedangkan $W[1,1] + W[1,2] = \infty + 7 = \infty$.

  Karena $W[1,2] \not> W[1,1] + W[1,2]$, maka harga $W[1,2]$ tetap.

- $W[5,4] = \infty$, sedangkan $W[5,1] + W[1,4] = 2 + 2 = 4$.

Karena $W[5,4] > W[5,1] + W[1,4]$, maka harga $W[5,4]$ diubah menjadi 4.

Ini berarti bahwa ada rute dari v_5 ke v_4 melalui v_1 yang mempunyai bobot lebih kecil (yaitu rute v_5 v_1 v_4 dengan jumlah bobot 4) dibandingkan dengan rute dari v_5 ke v_4 secara langsung (bobot = ∞ karena tidak ada rute dari v_5 ke v_4 secara langsung).

Dengan cara yang sama, harga $W[i, j]$ dihitung untuk setiap *i* dan *j*.

Didapatkan matiks:

$$W_1 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} \infty & 7 & \infty & 2 & \infty & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & \infty & \infty & \infty & \infty \\ 2 & 9 & 2 & 4 & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty \end{bmatrix} \end{array}$$

Iterasi untuk *k* = 2

Iterasi untuk *k* = 2 dilakukan dengan cara yang sama seperti iterasi untuk k = 1, hanya titik perantaranya adalah titik v_2.

Sebagai contoh:

- $W[6,5] = \infty$, sedangkan $W[6,2] + W[2,5] = 1 + 1 = 2$.

  Karena $W[6,5] > W[6,2] + W[2,5]$, maka harga $W[6,5]$ diubah menjadi 2.

  Ini berarti bahwa rute dari v_6 ke v_5 melalui v_2 $(v_6\ v_2\ v_5)$ lebih pendek dibandingkan rute dari v_6 ke v_5 secara langsung.

Proses yang sama dilakukan untuk semua harga *i* dan *j*.

Didapatkan:

$$W_2 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} \infty & 7 & 11 & 2 & 8 & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & \infty \\ 2 & 9 & 2 & 4 & 10 & \infty \\ \infty & 1 & 5 & \infty & 2 & \infty \end{bmatrix} \end{array}$$

Dengan cara yang sama, untuk *k = 3, 4, 5, 6*, diperoleh:

$$W_3 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} \infty & 7 & 11 & 2 & 8 & 14 \\ \infty & \infty & 4 & \infty & 1 & 7 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & 11 \\ 2 & 9 & 2 & 4 & 10 & 5 \\ \infty & 1 & 5 & \infty & 2 & 8 \end{bmatrix} \end{array}$$

$$W_4 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} \infty & 6 & 10 & 2 & 7 & 13 \\ \infty & \infty & 4 & \infty & 1 & 7 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & 11 \\ 2 & 8 & 2 & 4 & 9 & 5 \\ \infty & 1 & 5 & \infty & 2 & 8 \end{bmatrix} \end{array}$$

$$W_5 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} 9 & 6 & 9 & 2 & 7 & 12 \\ 3 & 9 & 3 & 5 & 1 & 6 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ 7 & 4 & 7 & 9 & 5 & 10 \\ 2 & 8 & 2 & 4 & 9 & 5 \\ 4 & 1 & 4 & 6 & 2 & 7 \end{bmatrix} \end{array}$$

$$W^* = W_6 = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} 9 & 6 & 9 & 2 & 7 & 12 \\ 3 & 7 & 3 & 5 & 1 & 6 \\ 7 & 4 & 7 & 9 & 5 & 3 \\ 7 & 4 & 7 & 9 & 5 & 10 \\ 2 & 6 & 2 & 4 & 7 & 5 \\ 4 & 1 & 4 & 6 & 2 & 7 \end{bmatrix} \end{array}$$

Iterasi selesai setelah iterasi untuk $k = n$. Jika pada W* ada W_{ij} dengan harga ∞ berarti tidak ada rute dari v_i ke v_j baik langsung maupun tidak langsung.

Untuk memudahkan dalam membaca rute terpendek yang diperoleh, tulis setiap sel matriks yang diganti dengan rute terpendek sel matriks yang baru.

1) Pada iterasi untuk $k = 1$

   $W[5,2] \rightarrow W[5,1] + W[1,2]$,

   $W[5,4] \rightarrow W[5,1] + W[1,4]$.

2) Pada iterasi untuk $k = 2$

   $W[1,3] \rightarrow W[1,2] + W[2,3]$,

   $W[1,5] \rightarrow W[1,2] + W[2,5]$,

   $W[4,3] \rightarrow W[4,2] + W[2,3]$,

   $W[4,5] \rightarrow W[4,2] + W[2,5]$,

   $W[5,5] \rightarrow W[5,2] + W[2,5]$,

   $W[6,5] \rightarrow W[6,2] + W[2,5]$.

3) Pada iterasi untuk $k = 3$

   $W[1,6] \rightarrow W[1,3] + W[3,6]$,

   $W[2,6] \rightarrow W[2,3] + W[3,6]$,

   $W[4,6] \rightarrow W[4,3] + W[3,6]$,

   $W[5,6] \rightarrow W[5,3] + W[3,6]$,

   $W[6,6] \rightarrow W[6,3] + W[3,6]$.

UNNES
UNIVERSITAS NEGERI SEMARANG

4) Pada iterasi untuk $k = 4$

$W[1,2] \rightarrow W[1,4] + W[4,2]$,

$W[1,3] \rightarrow W[1,4] + W[4,3]$,

$W[1,5] \rightarrow W[1,4] + W[4,5]$,

$W[1,6] \rightarrow W[1,4] + W[4,6]$,

$W[5,2] \rightarrow W[5,4] + W[4,2]$,

$W[5,5] \rightarrow W[5,4] + W[4,5]$.

5) Pada iterasi untuk $k = 5$

$W[1,1] \rightarrow W[1,5] + W[5,1]$,

$W[1,3] \rightarrow W[1,5] + W[5,3]$,

$W[1,6] \rightarrow W[1,5] + W[5,6]$,

$W[2,1] \rightarrow W[2,5] + W[5,1]$,

$W[2,2] \rightarrow W[2,5] + W[5,2]$,

$W[2,3] \rightarrow W[2,5] + W[5,3]$,

$W[2,4] \rightarrow W[2,5] + W[5,4]$,

$W[2,6] \rightarrow W[2,5] + W[5,6]$,

$W[4,1] \rightarrow W[4,5] + W[5,1]$,

$W[4,3] \rightarrow W[4,5] + W[5,3]$,

$W[4,4] \rightarrow W[4,5] + W[5,4]$,

$W[4,6] \rightarrow W[4,5] + W[5,6]$,

$W[6,1] \rightarrow W[6,5] + W[5,1]$,

$W[6,3] \rightarrow W[6,5] + W[5,3]$,

$W[6,4] \rightarrow W[6,5] + W[5,4]$,

UNNES
UNIVERSITAS NEGERI SEMARANG

$W[6,6] \rightarrow W[6,5] + W[5,6]$.

6) Pada iterasi untuk $k = 6$

$W[2,2] \rightarrow W[2,6] + W[6,2]$,

$W[3,1] \rightarrow W[3,6] + W[6,1]$,

$W[3,2] \rightarrow W[3,6] + W[6,2]$,

$W[3,3] \rightarrow W[3,6] + W[6,3]$,

$W[3,4] \rightarrow W[3,6] + W[6,4]$,

$W[3,5] \rightarrow W[3,6] + W[6,5]$,

$W[5,2] \rightarrow W[5,6] + W[6,2]$,

$W[5,5] \rightarrow W[5,6] + W[6,5]$.

Rute terpendek dari v_1 ke v_1 adalah 9 dengan rute terpendek yang dibaca dari iterasi terbesar ke iterasi terkecil sebagai berikut.

1) $W[1,1]$ pada iterasi dengan $k = 5$ diperoleh penurunan panjang rute dari ∞ ke 9.

Ini berarti rute $v_1 \rightarrow v_1$ diubah menjadi rute $v_1 \rightarrow v_5 \rightarrow v_1$.

Sehingga diperoleh rute terpendek $v_1 \rightarrow v_5 \rightarrow v_1$.

2) $W[1,5]$ pada iterasi dengan $k = 4$ diperoleh penurunan panjang rute dari 8 ke 7.

Ini berarti rute $v_1 \rightarrow v_5$ diubah menjadi rute $v_1 \rightarrow v_4 \rightarrow v_5$.

Sehingga diperoleh rute terpendek $v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_1$.

3) $W[4,5]$ pada iterasi dengan $k = 2$ diperoleh penurunan panjang rute dari ∞ ke 5.

Ini berarti rute $v_4 \rightarrow v_5$ diubah menjadi rute $v_4 \rightarrow v_2 \rightarrow v_5$.

Sehingga diperoleh rute terpendek $v_1 \rightarrow v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1$.

Jadi rute terpendek dari v_1 ke v_1 adalah 9 dengan rute terpendek $v_1 \rightarrow v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1$. Hasil rute terpendek dari semua titik yaitu sebagai berikut.

Tabel 2.2 Tabel hasil rute terpendek menggunakan Algoritma Floyd-Warshall

Titik Asal	Titik Akhir	Panjang Rute	Rute Terpendek
v_1	v_1	9	$v_1 \rightarrow v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1$
v_1	v_2	6	$v_1 \rightarrow v_4 \rightarrow v_2$
v_1	v_3	9	$v_1 \rightarrow v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3$
v_1	v_4	2	$v_1 \rightarrow v_4$
v_1	v_5	7	$v_1 \rightarrow v_4 \rightarrow v_2 \rightarrow v_5$
v_1	v_6	12	$v_1 \rightarrow v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$
v_2	v_1	3	$v_2 \rightarrow v_5 \rightarrow v_1$
v_2	v_2	7	$v_2 \rightarrow v_5 \rightarrow v_3 \rightarrow v_6 \rightarrow v_2$
v_2	v_3	3	$v_2 \rightarrow v_5 \rightarrow v_3$
v_2	v_4	5	$v_2 \rightarrow v_5 \rightarrow v_1 \rightarrow v_4$
v_2	v_5	1	$v_2 \rightarrow v_5$
v_2	v_6	6	$v_2 \rightarrow v_5 \rightarrow v_3 \rightarrow v_6$
v_3	v_1	7	$v_3 \rightarrow v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1$
v_3	v_2	4	$v_3 \rightarrow v_6 \rightarrow v_2$
v_3	v_3	7	$v_3 \rightarrow v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3$
v_3	v_4	9	$v_3 \rightarrow v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1 \rightarrow v_4$
v_3	v_5	5	$v_3 \rightarrow v_6 \rightarrow v_2 \rightarrow v_5$
v_3	v_6	3	$v_3 \rightarrow v_6$
v_4	v_1	7	$v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1$
v_4	v_2	4	$v_4 \rightarrow v_2$
v_4	v_3	7	$v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3$
v_4	v_4	9	$v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1 \rightarrow v_4$
v_4	v_5	5	$v_4 \rightarrow v_2 \rightarrow v_5$
v_4	v_6	10	$v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3 \rightarrow v_6$
v_5	v_1	2	$v_5 \rightarrow v_1$
v_5	v_2	6	$v_5 \rightarrow v_3 \rightarrow v_6 \rightarrow v_2$
v_5	v_3	2	$v_5 \rightarrow v_3$
v_5	v_4	4	$v_5 \rightarrow v_1 \rightarrow v_4$
v_5	v_5	7	$v_5 \rightarrow v_3 \rightarrow v_6 \rightarrow v_2 \rightarrow v_5$
v_5	v_6	5	$v_5 \rightarrow v_3 \rightarrow v_6$
v_6	v_1	4	$v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1$
v_6	v_2	1	$v_6 \rightarrow v_2$
v_6	v_3	4	$v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3$
v_6	v_4	6	$v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_1 \rightarrow v_4$
v_6	v_5	2	$v_6 \rightarrow v_2 \rightarrow v_5$
v_6	v_6	7	$v_6 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3 \rightarrow v_6$

2.11. Jalan

Jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada pada permukaan tanah, diatas permukaan tanah, di bawah permukaan tanah dan atau air, serta di atas permukaan air, kecuali jalan kereta api, jalan lori, dan jalan kabel (UU No. 38 tahun 2004 tentang Jalan).

Jalan dibedakan menjadi tiga macam.

1) Jalan Umum

Jalan umum adalah jalan yang diperuntukkan bagi lalu lintas umum.

2) Jalan Khusus

Jalan khusus adalah jalan yang dibangun oleh instansi, badan usaha, perseorangan, atau kelompok masyarakat untuk kepentingan sendiri.

3) Jalan Tol

Jalan tol adalah jalan umum yang merupakan bagian sistem jaringan jalan dan sebagai jalan nasional yang penggunanya diwajibkan membayar tol.

Jalan umum dikelompokkan menurut sistem, fungsi, status, dan kelas. Menurut sistem, jalan umum terdiri dari (1) sistem jaringan jalan primer, dan (2) sistem jaringan jalan sekunder. Menurut fungsi, jalan umum terdiri dari (1) jalan arteri, (2) jalan kolektor, (3) jalan lokal, dan (4) jalan lingkungan. Menurut status, jalan umum terdiri dari (1) jalan nasional, (2) jalan provinsi, (3) jalan kabupaten, (4) jalan kota, dan (5) jalan desa. Menurut kelas, jalan umum terdiri dari (1) jalan kelas I, (2) jalan kelas II, (3) jalan kelas IIA, (4) jalan kelas IIB, (5) jalan kelas IIC dan (6) jalan kelas III. Yang dibahas di sini adalah jalan umum menurut fungsi.

UNNES
UNIVERSITAS NEGERI SEMARANG

Jalan umum menurut fungsinya dikelompokkan menjadi:

1) **Jalan arteri**

   Jalan arteri merupakan jalan umum yang berfungsi melayani angkutan pengumpul atau pembagi dengan ciri perjalanan panjang rute jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara berdaya guna.

2) **Jalan kolektor**

   Jalan kolektor merupakan jalan umum yang berfungsi melayani angkutan pengumpul atau pembagi dengan ciri perjalanan panjang rute sedang, kecepatan rata-rata sedang, dan jumlah jalan masuk dibatasi.

3) **Jalan lokal**

   Jalan lokal merupakan jalan umum yang berfungsi melayani angkutan setempat dengan ciri perjalanan panjang rute dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk tidak dibatasi.

4) **Jalan lingkungan**

   Jalan lingkungan merupakan jalan umum yang berfungsi melayani angkutan lingkungan dengan ciri perjalanan panjang rute dekat dan kecepatan rata-rata rendah.

UNNES
UNIVERSITAS NEGERI SEMARANG

2.12. Volume Lalu Lintas

Berdasarkan MKJI Tahun 1997, volume lalu lintas adalah jumlah kendaraan yang melewati suatu penampang tertentu pada suatu ruas jalan tertentu dalam satuan waktu tertentu. Volume lalu lintas rata-rata adalah jumlah kendaraan rata-rata dihitung menurut suatu satuan waktu tertentu, bisa harian yang dikatakan

sebagai Volume Lalu Lintas Harian Rata-rata/ LHR yang dalam bahasa inggris disebut sebagai *Average Daily Traffic Volume* (ADT) atau Volume Lalu Lintas Harian Rata-rata Tahunan yang dalam bahasa inggris disebut sebagai *Annual Average Daily Traffic Volume* (AADT).

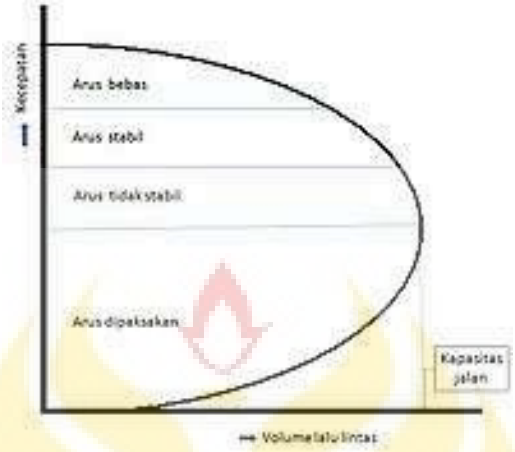
2.13. Kapasitas Jalan

Menurut Yunianta (2006), kapasitas jalan adalah kemampuan ruas jalan untuk menampung arus atau volume lalu lintas yang ideal dalam satuan waktu tertentu, dinyatakan dalam jumlah kendaraan yang melewati potongan jalan tertentu dalam satu jam (kend/jam) atau dengan mempertimbangkan berbagai jenis kendaraan yang melalui suatu jalan digunakan satuan mobil penumpang sebagai satuan kendaraan dalam hitungan kapasitas maka kapasitas menggunakan satuan satuan mobil penumpang per jam atau (smp/jam).

Pada saat arus rendah kecepatan lalu lintas kendaraan bebas tidak ada gangguan dari kendaraan lain, semakin banyak kendaraan yang melewati ruas jalan, kecepatan akan semakin turun sampai suatu saat tidak bisa lagi arus/volume lalu lintas bertambah, disinilah kapasitas terjadi. Setelah itu arus akan berkurang terus dalam kondisi arus yang dipaksakan sampai suatu saat kondisi macet total, arus tidak bergerak dengan kepadatan tinggi.

UNNES
UNIVERSITAS NEGERI SEMARANG

Grafik yang menunjukkan bahwa kapasitas jalan merupakan arus maksimum adalah sebagai berikut.



Gambar 2.9 Grafik Kapasitas Jalan

2.14. Tingkat Pelayanan (V/C Ratio)

Tingkat pelayanan berdasarkan KM 14 Tahun 2006 tentang Manajemen dan Rekayasa Lalu Lintas di Jalan diklasifikasikan menjadi tingkat pelayanan A, tingkat pelayanan B, tingkat pelayanan C, tingkat pelayanan D, tingkat pelayanan E, tingkat pelayanan F.

Tingkat pelayanan ini bisa dihitung dengan cara membagi volume lalu lintas dengan kapasitas jalan dengan kata lain V/C ratio merupakan perbandingan antara volume ruas jalan pada jam tersibuk dengan kapasitas total jalan tersebut. Besarnya tingkat pelayanan ini yang menentukan suatu jalan termasuk dalam tingkat pelayanan yang mana.

UNNES
UNIVERSITAS NEGERI SEMARANG

Tabel 2.3 Tabel karakteristik tingkat pelayanan beserta batas lingkupnya

Tingkat Pelayanan	Karakteristik-karakteristik	Batas Lingkup (Volume/Kapasitas)
A	1. Kondisi arus bebas dengan kecepatan tinggi, dan volume lalu lintas rendah 2. Kepadatan lalu lintas sangat rendah dengan kecepatan yang dapat dikendalikan oleh pengemudi 3. Pengemudi dapat memilih kecepatan yang diinginkannya tanpa hambatan	0,00 – 0,24
B	1. Kondisi arus stabil dengan volume lalu lintas sedang dan kecepatan mulai dibatasi oleh kondidi lalu lintas 2. Kepadatan lalu lintas rendah hambatan internal lalu lintas belum mempengaruhi kecepatan 3. Pengemudi memiliki kebebasan yang cukup untuk memilih kecepatan dan lajur jalan yang digunakan	0,25 – 0,44
C	1. Kondisi arus stabil tetapi kecepatan dan pergerakan kendaraan dikendalikan oleh volume lalu lintas yang lebih tinggi 2. Kepadatan lalu lintas sedang karena hambatan internal lalu lintas meningkat 3. Pengemudi memiliki keterbatasan untuk memilih kecepatan, pindah lajur atau mendahului	0,45 – 0,74
D	1. Kondisi arus mendekati tidak stabil dengan volume lalu lintas tinggi dan kecepatan masih ditolerir namun sangat terpengaruh oleh perubahan kondisi arus 2. Kepadatan lalu lintas sedang namun fluktuasi volume lalu lintas dan hambatan temporer dapat menyebabkan penurunan kecepatan yang besar 3. Pengemudi memiliki kebebasan yang sangat terbatas dalam menjalankan kendaraan, kenyamanan rendah, tetapi kondisi masih dapat ditolerir untuk waktu yang singkat	0,75 – 0,84
E	1. Kondisi arus tidak stabil dengan Volume lalu lintas mendekati kapasitas jalan dan	0,85 – 1,00

	kecepatan rendah 2. Kepadatan lalu lintas tinggi karena hambatan internal lalu lintas tinggi 3. Pengemudi mulai merasakan kemacetan-kemacetan durasi pendek	
F	1. Kondisi arus tertahan dan terjadi antrian kendaraan yang panjang 2. Kepadatan lalu lintas sangat tinggi dan volume rendah serta terjadi kemacetan untuk durasi yang cukup lama 3. Antrian yang panjang, kecepatan maupun volume turun sampai 0	> 1,00

2.15. Kepadatan

Menurut Tampubolon (2012), kepadatan diartikan sebagai jumlah kendaraan yang ada pada satu ruas jalan raya atau rute biasanya dinyatakan sebagai jumlah kendaraan per kilometer atau satuan mobil penumpang per kilometer (smp/km). Jika panjang ruas yang diamati adalah L, dan terdapat N kendaraan, maka kepadatan yang dilambangkan dengan "*k*" dapat dihitung sebagai berikut.

$$k = \frac{N}{L} \quad \text{..... (2.4)}$$

Kepadatan sukar diukur secara langsung, karena diperlukan titik ketinggian tertentu yang dapat mengamati jumlah kendaraan dalam panjang ruas jalan tertentu, sehingga besarnya ditentukan dari dua parameter volume dan kecepatan yang mempunyai hubungan sebagai berikut.

$$k = \frac{Volume\ lalu\ lintas}{Kecepatan} \quad \text{..... (2.5)}$$

Diuraikan menjadi:

$$k = \frac{Volume\ lalu\ lintas}{\frac{Panjang\ ruas\ jalan}{Waktu\ perjalanan}} \quad \text{..... (2.6)}$$

Atau bisa ditulis dengan:

$$k = \frac{Volume\ lalu\ lintas \times Waktu\ perjalanan}{Panjang\ ruas\ jalan} \quad \text{..... (2.7)}$$

Diperoleh persamaan:

$$Waktu\ perjalanan = \frac{k \times Panjang\ ruas\ jalan}{Volume\ lalu\ lintas} \quad \text{..... (2.8)}$$

Berarti waktu perjalanan merupakan hasil dari perkalian antara kepadatan dan panjang ruas jalan dibagi volume lalu lintas (Dishubkominfo, 2016:216).

2.16. Kemacetan

Kemacetan adalah situasi atau keadaan tersendatnya atau bahkan terhentinya lalu lintas yang disebabkan oleh banyaknya jumlah kendaraan melebihi kapasitas jalan. Kemacetan banyak terjadi di kota-kota besar, terutama yang tidak mempunyai transportasi publik yang baik atau memadai ataupun juga tidak seimbangnya kebutuhan jalan dengan kepadatan penduduk (Tampubolon, 2012).

2.16.1. Penyebab kemacetan

Kemacetan dapat terjadi karena beberapa alasan, diantaranya sebagai berikut.

1. Arus yang melewati jalan telah melampaui kapasitas jalan
2. Terjadi kecelakaan lalu lintas

3. Terjadi bencana alam seperti banjir, tanah longsor
4. Terjadinya kebakaran
5. Adanya perbaikan jalan
6. Adanya rumah-rumah kumuh/ bangunan liar
7. Adanya kendaraan keluar-masuk, kendaraan yang berhenti atau parkir sembarangan serta adanya pedagang asongan
8. Adanya pemakai jalan yang tidak tahu aturan lalu lintas
9. Adanya parkir liar dari sebuah kegiatan
10. Adanya pasar tumpah yang secara tidak langsung memakan badan jalan
11. Adanya tawuran pelajar ataupun aksi demo
12. Kemacetan lalu lintas di perrute sebidang karena kereta api yang lewat
13. Kemacetan lalu lintas yang disebabkan kepanikan
14. Pengaturan lampu lalu lintas yang bersifat kaku yang tidak mengikuti tinggi rendahnya arus lalu lintas

2.16.2. Dampak negatif kemacetan

Kemacetan lalu lintas memberikan dampak negatif yang besar, antara lain:

1. Kerugian waktu, karena kecepatan perjalanan rendah
2. Pemborosan energi, karena pada kecepatan rendah konsumsi bahan bakar rendah tetapi memakan waktu yang lama
3. Keausan kendaraan lebih tinggi, karena waktu yang lebih lama untuk panjang rute yang pendek, radiator tidak berfungsi dengan baik, dan penggunaan rem yang lebih tinggi

UNNES
UNIVERSITAS NEGERI SEMARANG

4. Meningkatkan polusi udara karena kecepatan rendah konsumsi energi lebih tinggi dan mesin tidak beroperasi pada kondisi normal
5. Meningkatkan stress pengguna jalan
6. Mengganggu kelancaran kendaraan darurat seperti ambulans dan pemadam kebakaran dalam menjalankan tugasnya

2.16.3. Pemecahan permasalahan kemacetan

Yang bisa dilakukan untuk memecahkan permasalahan kemacetan lalu lintas yaitu ebagai berikut.

1. Peningkatan kapasitas jalan
2. Keberpihakan kepada angkutan umum
3. Pembatasan kendaraan pribadi

UNNES
UNIVERSITAS NEGERI SEMARANG

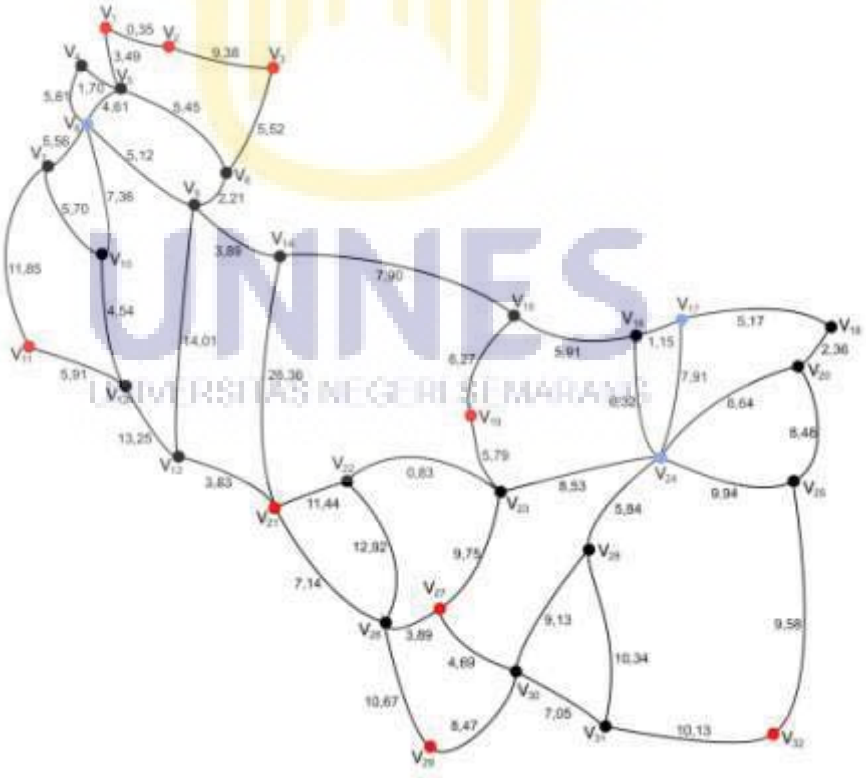
BAB 5

PENUTUP

5.1. Simpulan

Berdasarkan hasil penelitian dan pembahasan tentang penerapan algoritma Dijkstra dan Floyd-Warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang, dapat ditarik kesimpulan sebagai berikut.

1. Rute tempat wisata di Batang dapat direpresentasikan sebagai graf berbobot berikut.



Gambar 5.1 Graf Berbobot Rute Tempat Wisata di Batang

2. Penyelesaian dari penerapan algoritma Dijkstra dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang yaitu sebagai berikut.

Tabel 5.1 Tabel hasil penerapan Algoritma Dijkstra dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang

Stasiun/ Tempat Terminal Wisata	Stasiun Batang Baru	Terminal Banyuputih	Terminal Limpung
Pantai Sigandu	$v_6 \rightarrow v_5 \rightarrow v_1$ (8,10 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1$ (30 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1$ (35,17 menit)
Batang Dolphins Center	$v_6 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (8,45 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (30,35 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (35,52 menit)
Pantai Ujung Negoro	$v_6 \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (12,85 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (26,58 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (31,75 menit)
Tubing Pandansari	$v_6 \rightarrow v_7 \rightarrow v_{11}$ (17,41 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_6 \rightarrow v_7 \rightarrow v_{11}$ (41,38 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21} \rightarrow v_{13} \rightarrow v_{12} \rightarrow v_{11}$ (43,79 menit)
Curug Gombong	$v_6 \rightarrow v_9 \rightarrow v_{14} \rightarrow v_{15} \rightarrow v_{19}$ (23,18 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{19}$ (13,33 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{19}$ (14,32 menit)
Bandar Ecopark	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21}$ (22,96 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21}$ (28,27 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21}$ (20,80 menit)
Wisata Agro Selopajang Timur	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21} \rightarrow v_{26} \rightarrow v_{27}$ (33,99 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{27}$ (25,75 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{27}$ (18,28 menit)
Agrowisata Teh Pagilaran	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21} \rightarrow$	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{28} \rightarrow$	$v_{24} \rightarrow v_{28} \rightarrow v_{30} \rightarrow v_{29}$

	$v_{26} \rightarrow v_{29}$ (40,77 menit)	$v_{30} \rightarrow v_{29}$ (30,91 menit)	(23,44 menit)
Pemandian Air Hangat Sangubanyu	$v_6 \rightarrow v_9 \rightarrow v_{14} \rightarrow v_{15} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{25} \rightarrow v_{32}$ (48,67 menit)	$v_{17} \rightarrow v_{18} \rightarrow v_{20} \rightarrow v_{25} \rightarrow v_{32}$ (25,59 menit)	$v_{24} \rightarrow v_{25} \rightarrow v_{32}$ (19,52 menit)

3. Penyelesaian dari penerapan algoritma Floyd-Warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang yaitu sebagai berikut.

Tabel 5.2 Tabel hasil penerapan Floyd-Warshall dalam menentukan rute terpendek dari stasiun/terminal di Batang menuju ke tempat wisata di Batang

Stasiun/ Terminal / Tempat Wisata	Stasiun Batang Baru	Terminal Banyuputih	Terminal Limpung
Pantai Sigandu	$v_6 \rightarrow v_5 \rightarrow v_1$ (8,10 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1$ (30 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1$ (35,17 menit)
Batang Dolphins Center	$v_6 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (8,45 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (30,35 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (35,52 menit)
Pantai Ujung Negoro	$v_6 \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (12,85 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (26,58 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (31,75 menit)
Tubing Pandansari	$v_6 \rightarrow v_7 \rightarrow v_{11}$ (17,41 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21} \rightarrow v_{13} \rightarrow v_{12} \rightarrow v_{11}$ (51,26 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21} \rightarrow v_{13} \rightarrow v_{12} \rightarrow v_{11}$ (43,79 menit)
Curug Gombong	$v_6 \rightarrow v_9 \rightarrow v_{14} \rightarrow v_{15} \rightarrow v_{19}$ (23,18 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{19}$ (13,33 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{19}$ (14,32 menit)

Bandar Ecopark	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21}$ (22,96 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{27} \rightarrow v_{26} \rightarrow v_{21}$ (36,78 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21}$ (20,80 menit)
Wisata Agro Selopajang Timur	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21} \rightarrow v_{26} \rightarrow v_{27}$ (33,99 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{27}$ (25,75 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{27}$ (18,28 menit)
Agrowisata Teh Pagilaran	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21} \rightarrow v_{26} \rightarrow v_{29}$ (40,77 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{28} \rightarrow v_{30} \rightarrow v_{29}$ (30,91 menit)	$v_{24} \rightarrow v_{28} \rightarrow v_{30} \rightarrow v_{29}$ (23,44 menit)
Pemandian Air Hangat Sangubanyu	$v_6 \rightarrow v_9 \rightarrow v_{14} \rightarrow v_{15} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{25} \rightarrow v_{32}$ (48,67 menit)	$v_{17} \rightarrow v_{18} \rightarrow v_{20} \rightarrow v_{25} \rightarrow v_{32}$ (25,59 menit)	$v_{24} \rightarrow v_{25} \rightarrow v_{32}$ (19,52 menit)

4. Rute terpendek yang direkomendasikan yaitu sebagai berikut.

Tabel 5.3 Tabel rute terpendek yang direkomendasikan dari stasiun/terminal di Batang menuju ke tempat wisata di Batang

Stasiun/ Terminal / Tempat Wisata	Stasiun Batang Baru	Terminal Banyuputih	Terminal Limpung
Pantai Sigandu	$v_6 \rightarrow v_5 \rightarrow v_1$ (8,10 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1$ (30 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1$ (35,17 menit)
Batang Dolphins Center	$v_6 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (8,45 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (30,35 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_5 \rightarrow v_1 \rightarrow v_2$ (35,52 menit)
Pantai Ujung Negoro	$v_6 \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (12,85 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (26,58 menit)	$v_{24} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_8 \rightarrow v_3$ (31,75 menit)
Tubing Pandansari	$v_6 \rightarrow v_7 \rightarrow v_{11}$	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_9 \rightarrow v_6 \rightarrow v_7 \rightarrow$	$v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21} \rightarrow v_{13} \rightarrow$

	(17,41 menit)	v_{11} (41,38 menit)	$v_{12} \rightarrow v_{11}$ (43,79 menit)
Curug Gombong	$v_6 \rightarrow v_9 \rightarrow v_{14} \rightarrow v_{15} \rightarrow v_{19}$ (23,18 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{15} \rightarrow v_{19}$ (13,33 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{19}$ (14,32 menit)
Bandar Ecopark	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21}$ (22,96 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21}$ (28,27 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{21}$ (20,80 menit)
Wisata Agro Selopajang Timur	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21} \rightarrow v_{26} \rightarrow v_{27}$ (33,99 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{23} \rightarrow v_{27}$ (25,75 menit)	$v_{24} \rightarrow v_{23} \rightarrow v_{27}$ (18,28 menit)
Agrowisata Teh Pagilaran	$v_6 \rightarrow v_9 \rightarrow v_{13} \rightarrow v_{21} \rightarrow v_{26} \rightarrow v_{29}$ (40,77 menit)	$v_{17} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{28} \rightarrow v_{30} \rightarrow v_{29}$ (30,91 menit)	$v_{24} \rightarrow v_{28} \rightarrow v_{30} \rightarrow v_{29}$ (23,44 menit)
Pemandian Air Hangat Sangubanyu	$v_6 \rightarrow v_9 \rightarrow v_{14} \rightarrow v_{15} \rightarrow v_{16} \rightarrow v_{24} \rightarrow v_{25} \rightarrow v_{32}$ (48,67 menit)	$v_{17} \rightarrow v_{18} \rightarrow v_{20} \rightarrow v_{25} \rightarrow v_{32}$ (25,59 menit)	$v_{24} \rightarrow v_{25} \rightarrow v_{32}$ (19,52 menit)

5.2. Saran

Berdasarkan simpulan di atas, saran yang dapat disampaikan penulis yaitu sebagai berikut.

1. Hasil penelitian ini dapat digunakan sebagai bahan informasi tentang rute terpendek menuju tempat wisata di Batang yang dapat digunakan untuk melengkapi situs web yang dibuat oleh Dinas Kebudayaan dan Pariwisata Kabupaten Batang serta sebagai bahan acuan bagi para wisatawan yang menginginkan rute yang paling efisien untuk menuju tempat wisata di Batang.

UNNES
UNIVERSITAS NEGERI SEMARANG

2. Paket wisata dapat digunakan sebagai bahan informasi bagi para wisatawan tentang tempat wisata yang dikelompokan sehingga mereka dapat singgah di tempat wisata yang dilewati sebelum sampai ke tempat wisata tujuan.

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR PUSTAKA

Aini, A. & A. Salehipour. 2012. Speeding Up The Floyd-Warshall Algorithm for The Cycled Shortest Path Problem. Iran: Elsevier Ltd. *Journal of Applied Mathematics Letters 25 (2012) 1-5*. Tersedia di http://www.sciencedirect.com/science/article/pii/S0893965911002928 [4 Maret 2016].

Bell, E. T. 1952. *Mathematics: Queen and Servant of Science*. London: G. Bell & Sons, Ltd.

Budayasa, I Ketut. 2007. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.

Dimyati, T. & A. Dimyati. 2004. *Operations Research: Model-model Pengambilan Keputusan.* Bandung: Sinar Baru Algensindo.

Disbudpar. 2016. *Guide Tourisme Batang – Indonesia*. Batang: Dinas Kebudayaan dan Pariwisata Kabupaten Batang.

Dishubkominfo. 2016. *Survey Database Tingkat Pelayanan Jalan*. Batang: Dinas Perhubungan Komunikasi dan Informatika Kabupaten Batang.

Faro, A. & D. Giordano. 2016. Algorithm to Find Shortest and Alternative Path in Free Flow and Congested Traffic Regimes. Italy: Elsevier B.V. *Journal of Electrical, Electrinics and Computer Engineering 73 (2016) 1-29*. Tersedia di http://www.sciencedirect.com/science/article/pii/S0968090X16301760 [29 November 2016].

Galan-Garcia, J. L. *et al*. 2014. A New Probabilistic of Dijkstra's Algorithm to Simulate More Realistic Traffict Flow in A Smart City. Spain: Elsevier Inc. *Journal of Applied Mathematics and Computation xxx (2014) xxx-xxx*. Tersedia di http://www.sciencedirect.com/science/article/pii/S0096300314016087 [4 Maret 2016].

Hougardy, S. 2010. The Floyd-Warshall Algorithm on Graph With Negative Cycles. Germany: Elsevier B. V. *Journal of Information Processing Letters 110 (2010) 279-281*. Tersedia di http://www.sciencedirect.com/science/article/pii/S002001901000027X [4 Maret 2016].

Kreyszig, E. 1993. *Matematika Teknik Lanjutan*. Jakarta: Gramedia Pustaka Utama.

Mardlootillah, H. I., A. Suyitno, & F. Y. Arini. 2015. Simulasi Algoritma Dijkstra dalam Menangani Masalah Lintasan Terpendek pada Graf Menggunakan Visual Basic. Indonesia: *UNNES Journal of Mathematics 4 (2) (2015)*. Tersedia di http://journal.unnes.ac.id/sju/index.php/ujm/article/view/9349 [11 Februari 2017]

MKJI. 1997. *Manual Kapasitas Jalan Indonesia (MKJI).* Jakarta: Direktorat Jendral Bina Marga Departemen Pekerjaan Umum.

Munir, R. 2010. *Matematika Diskrit*. Bandung: Informatika.

Novandi, R. A. 2007. *Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path).* Bandung: Institut Teknologi Bandung. Tersedia di http://webmail.informatika.org/~rinaldi/Stmik/2006-2007/Makalah_2007/MakalahSTMIK2007-021.pdf [20 Februari 2016].

Pandey, H. M. 2008. *Design Analysis and Algoritms*. India: University Science Press New Delhi.

Peta Wisata dan Potensi Wisata Kabupaten Batang. Tersedia di http://www.batanggallery.or.id/p/peta-wisata-kabupaten-batang.html [4 Mei 2016].

Peyer, Sven., D. Rautenbach, & J. Vygen. 2009. A Generalization of Dijkstra's Shortest Path Algorithm Applications to VLSI routing. Germany: Elsevier B.V. *Journal of Discrete Algorithm 7(2009) 377-390*. Tersedia di http://www.sciencedirect.com/science/article/pii/S1570866709000653 [4 Maret 2016].

Siang, J. J. 2004. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi Offset.

Situs Resmi Dinas Kebudayaan dan Pariwisata Kabupaten Batang. Tersedia di http://pariwisata.batangkab.go.id/ [14 Maret 2016].

Tampubolon, R. E. 2012. *Analisis Pengaruh Pasar Tradisional terhadap Kinerja Ruas Jalan*. Medan: Universitas Sumatera Utara. Tersedia di http://202.0.107.5/index.php/jts/article/view/6023/2517 [19 September 2016].

Yunianta, A. 2006. *Pengaruh Manuver Kendaraan Parkir Badan Jalan terhadap Karakteristik Arus Lalu Lintas Di Jalan Diponegoro Yogyakarta*. Semarang: Universitas Diponegoro.

PEMERINTAH KABUPATEN BATANG
DINAS KEBUDAYAAN DAN PARIWISATA
Jalan RA. Kartini No. 1 Telp/Fax (0285) 391141 Batang 51215

SURAT KETERANGAN
Nomor : 556/ 855 /2016

Yang bertanda tangan di bawah ini Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Batang, dengan ini menerangkan bahwa :

Nama	: LENI MARLINA
Pekerjaan	: Mahasiswa UNNES Semarang
Alamat	: Desa Boja RT.07 RW.04 Dk. Mranggen Kecamatan Tersono Kabupaten Batang

Benar – benar telah melaksanakan penelitian dan pengambilan data guna menyusun Skripsi dengan judul "Penerapan Algoritma di JKRSTRA dan Floyd-Warsall Untuk Menentukan Rute Terpendek Tempat Wisata di Batang" di lingkungan Dinas Kebudayaan dan Pariwisata Kabupaten Batang dari tanggal 8 s.d 12 Agustus 2016

Demikian, Surat Keterangan ini dibuat untuk digunakan sebagaimana mestinya.

Dikeluarkan di : Batang
pada tanggal : 15 Agustus 2016

An. KEPALA DINAS
KEBUDAYAAN DAN PARIWISATA
KABUPATEN BATANG
Sekretaris

PEMERINTAH KABUPATEN DISBUDPAR

SURRAYITNO, S.Kar,M.Si
NIP 19640412 198903 1 019

Tembusan : disampaikan kepada Yth.
1. Bapak Bupati Batang (Sebagai Laporan);
2. Asisten II Sekda Kab. Batang;
3. Arsip.