PERBANDINGAN ALGORITMA GREEDY DAN ALGORITMA A* PADA PENYELESAIAN *TRAVELLING SALESMAN PROBLEM*

(Studi Kasus: Pendistribusian Air Minum Isi Ulang di *Home Industry* Rizqua Kabupaten Pekalongan)

Skripsi

disajikan sebagai salah satu syarat untuk

memperoleh gelar Sarjana Sains

Program Studi Matematika

oleh

Mualif Akhyar

4111412027

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2017

PERNYATAAN

Saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam skripsi ini dan disebutkan dalam daftar pustaka.

Semarang, Desember 2017

METERAI TEMPEL
AD0BBAEF760197405
6000
ENAM RIBU RUPIAH

Mualif Akhyar
NIM. 4111412027

UNNES
UNIVERSITAS NEGERI SEMARANG

ii

PENGESAHAN

Skripsi yang berjudul

Perbandingan Algoritma Greedy dan Algoritma A* pada Penyelesaian *Travelling Salesman Problem.*

Disusun oleh

Mualif Akhyar

4111412027

Telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal 24 Oktober 2017.

Panitia,

Ketua

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

NIP. 196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si.

NIP. 196807221993031005

Ketua Penguji

Drs. Amin Suyitno, M.Pd.

NIP. 195206041976121001

Anggota Penguji

Pembimbing Utama

Dr. Mulyono, M.Si.

NIP. 197009021997021001

Anggota Penguji

Pembimbing Pendamping

Dr. Rochmad, M.Si.

NIP. 195711161987011001

MOTTO DAN PERSEMBAHAN

Motto

- Berani mengambil tantangan harus berani menerima segala resiko yang ada dan harus bisa menyelesaikannya. Bagi saya skripsi adalah sebuah tantangan untuk seorang mahasiswa yang menantang bagaimana mengatur waktu dan melatih kesabaran.
- Sesuatu yang belum dimulai memang berat, tetapi setelah memulai dan menikmati sesuatu itu akan menjadi lebih mudah.
- Prioritas hidup bukan hanya hobi, melainkan sesuatu yang dianggap penting untuk masa depan.
- Mendekatlah dan ceritakan keluh kesahmu kepada Sang Pencipta, maka sesuatu yang akan dikerjakan menjadi ringan.

Persembahan

Skripsi ini saya persembahkan untuk:

- Bapak, Ibu, Adik, dan Keluarga Besar di Rumah
- Keluarga Besar The Mathematics Adventure Team
- Teman-teman Matematika Angkatan 2012
- Teman & Sahabat Semuanya

UNNES
UNIVERSITAS NEGERI SEMARANG

KATA PENGANTAR

Alhamdulilah puja dan puji syukur penulis ke hadirat Allah SWT yang telah memberikan hidayah dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Perbandingan Algoritma Greedy dan Algoritma A* pada Penyelesaian *Travelling Salesman Problem*".

Skripsi ini dapat tersusun dengan baik berkat bantuan dan bimbingan beberapa pihak. Oleh karena itu, penulis menyampaikan terimakasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang, yang telah memberikan kesempatan kepada penulis untuk menyelesaikan studi strata 1 di jurusan Matematika FMIPA UNNES.
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 Universitas Negeri Semarang, yang telah memberikan izin untuk melakukan penelitian.
4. Dr. Mulyono, M.Si., selaku pembimbing I, yang telah menuntun, memberikan arahan dan bimbingan dalam penyelesaian skripsi ini.
5. Dr. Rochmad, M.Si,. selaku pembimbing II, yang telah menuntun, memberikan arahan dan bimbingan dalam penyelesaian skripsi ini.
6. Drs. Amin Suyitno, M.Pd., selaku penguji, yang telah berkenan untuk menguji skripsi ini.
7. Dra. Kristina Wijayanti, M.S., selaku dosen wali yang telah memberikan arahan dan bimbingan selama masa kuliah.

8. Kabul Sigit Setianto, selaku pemilik *home industry* air minum isi ulang Rizqua di Kabupaten Pekalongan yang telah memberikan izin untuk melakukan penelitian.
9. Bapak Marwoto, Ibu Susi Ida Hartini, Adik Firman Hidayat, Mbah uti, Mbah kung, Om, Tante yang selalu mendoakan dan menjadi motivasi dalam menyelesaikan skripsi ini.
10. Teman rumah Noki, Tahu, Adit, Bayu, Anggar, Ragil, Dimas yang telah mendukung saya menyelesaikan skripsi ini.
11. Teman Kos Erie, Gilang, Bayu E, Dwi, Rahmad, Adi, Om Sun, Papang, Pak Nying, Huda yang telah mendukung saya menyelesaikan skripsi ini.
12. Keluarga Besar The Mathematics Adventure Team yang telah memberikan banyak pelajaran hidup.
13. Teman-teman Matematika angkatan 2012, yang selama ini menemani dalam perkuliahan.
14. Adib, Dwi, Arif, Lusy, Kintan, Gina yang telah mendukung saya dalam menyelesaikan skripsi ini.
15. Lusy Rositawati yang telah sabar membantu saya dalam segala masalah dan memberikan semangat dalam menyelesaikan skripsi ini.
16. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam penyelesaian skripsi ini.

Hanya ucapan terima kasih dan doa, semoga apa yang telah diberikan tercatat sebagai amal baik dan mendapatkan balasan dari Allah SWT.

UNNES
UNIVERSITAS NEGERI SEMARANG

Semoga skripsi ini bisa membawa manfaat bagi penulis sendiri khususnya dan bagi para pembaca pada umumnya.

Semarang, Desember 2017

Penulis

UNNES
UNIVERSITAS NEGERI SEMARANG

ABSTRAK

Akhyar, Mualif. 2017. *Perbandingan Algoritma Greedy dan Algoritma A* pada Penyelesaian Travelling Salesman Problem*. Skripsi. Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Pertama Dr. Mulyono, M.Si. dan Pembimbing Kedua Dr. Rochmad, M.Si.

Kata Kunci: *Traveliing Salesman Problem,* Pendistribusian Barang, Algoritma Greedy, Algoritma A*, *Javascript*.

Teori graf merupakan cabang dari matematika yang sebenarnya sudah ada sejak lebih dari dua ratus tahun silam. Menemukan rute terpendek adalah usaha untuk mencari rute yang paling pendek jaraknya dari posisi awal hingga akhir dengan nilai yang paling kecil dibandingkan dengan seluruh rute yang ada. Tujuan dari penelitian ini yakni untuk mengetahui bagaimana penerapan algoritma Greedy dan algoritma A* untuk mencari rute terpendek pendistribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan dan mengetahui apakah implementasi algoritma Greedy dan algoritma A* dapat berpengaruh terhadap *home industry* Rizqua di Kabupaten Pekalongan.

Metode penelitian yang digunakan yaitu identifikasi permasalahan, investigasi awal yang meliputi studi literatur, observasi, menetukan masalah, dan tujuan, persiapan penelitian yang meliputi perijinan dan pengambilan data di lapangan, penyelesaian yang meliputi perencanaan, pengolahan data dan analisis data. Hasil penelitian berdasarkan algoritma A* untuk mengetahui jarak yang dapat dilalui agar memperoleh jarak terpendek tanpa agar memperoleh jarak terpendek tanpa memperdulikan kondisi kepadatan lalu lintas diperoleh panjang rute 19,55 Km. Sedangkan algoritma A* untuk mengetahui kepadatan lalu lintas agar memperoleh efisiensi waktu tempuhnya dengan memperdulikan kondisi jaraknya diperoleh panjang rute 27,80 Km. Berdasarkan algoritma Greedy untuk mengetahui jarak yang dapat dilalui agar memperoleh jarak terpendek tanpa memperdulikan kondisi kepadatan lalu lintas diperoleh panjang rute 23,95 Km. Sedangkan algoritma Greedy untuk mengetahui kepadatan lalu lintas agar memperoleh efisiensi waktu tempuhnya dengan mempedulikan kondisi jarak diperoleh hasil panjang rute 28,7 Km. Rute yang dihasilkan algoritma A* dapat menyelesaikan masalah pendistribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan, karena panjang rute yang dihasilkan lebih pendek daripada rute dari *home industry* tersebut.

DAFTAR ISI

HALAMAN JUDUL i

PERNYATAAN ii

PENGESAHAN iii

MOTTO DAN PERSEMBAHAN iv

KATA PENGANTAR v

ABSTRAK viii

DAFTAR ISI ix

DAFTAR TABEL xv

DAFTAR GAMBAR xvi

DAFTAR LAMPIRAN xviii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 5

1.3 Batasan Masalah 6

1.4 Tujuan Penelitian 6

1.5 Manfaat Penelitian 6

1.6 Sistematika Penulisan 7

BAB II TINJAUAN PUSTAKA 9

2.1 *Program Linear* 9

2.2 Riset Operasi 9

2.3 *Travelling Salesman Problem* (TSP) 11

2.3.1 Konsep *Travelling Salesman Problem* 12

UNNES
UNIVERSITAS NEGERI SEMARANG

2.4 Rute Terpendek 13

2.5 Teori Graf 13

2.5.1 Keterhubungan Pada Graf 14

2.5.1.1 Jalan (*Walk*) 14

2.5.1.2 Jejak (*Trail*) 15

2.5.1.3 Lintasan (*Path*) 15

2.5.1.4 Sikel 15

2.5.1.5 Lintasan Hamilton dan Sirkuit Hamilton 16

2.5.2 Jenis-Jenis Graf 17

2.5.2.1 Berdasarkan Ada Tidaknya Gelung Atau Sisi Ganda 17

2.5.2.1.1 Graf Sederhana (*Simple Graph*) 17

2.5.2.1.2 Graf Tidak Sederhana (*Unsimple Graph*) ... 17

2.5.2.2 Berdasarkan Orientasi Arah Pada Sisi 18

2.5.2.2.1 Graf Berarah (*Directed Graph* atau *Digraph*)18

2.5.2.2.2 Graf Tidak Berarah (*UndirectedGraph*) 18

2.5.2.3 Graf Lengkap (Graf Komplit) 19

2.5.2.4 Graf Kosong 19

2.5.2.5 Graf Bipartisi 20

2.5.2.6 Graf Terhubung (*Connected Graph*) & Graf Tak Terhubung (*Disconnected Graph*) 20

2.5.2.7 Sub Graf 21

2.5.2.8 Graf Euler dan Graf Semi-Euler 21

2.5.2.9 Graf Berlabel 22

UNNES
UNIVERSITAS NEGERI SEMARANG

2.5.3 Terminologi Graf 22
2.5.3.1 Ketetanggaan 22
2.5.3.2 Derajat 22
2.5.3.3 Bersisian 23
2.6 Metode Heuristik 23
2.7 Algoritma 24
2.8 Algoritma Greedy 24
2.9 Algoritma A* 27
2.10 Kapasitas Jalan 29
2.11 Arus Lalu Lintas 29
2.12 Kecepatan 30
2.13 Volume Jalan 30
2.14 Kepadatan Jalan 30
2.15 *Javascript* 31
2.15.1 Mengenal *Javascipt* 31
2.15.2 Perbedaan *Java* dengan *Javascript* 31
2.15.3 Pemakaian *Javascript* 32
2.15.4 *Javascript Grammar* 32
2.15.5 Memanipulasi *Window* 33
BAB III METODE PENELITIAN 35
3.1 Tahap Identifikasi Permasalahan 37
3.2 Investigasi Awal 37
3.2.1 Studi Literatur 37

UNNES
UNIVERSITAS NEGERI SEMARANG

3.2.2 Observasi 37

3.2.3 Menentukan Masalah..... 37

3.2.4 Tujuan..... 38

3.3 Persiapan Penelitian..... 38

3.3.1 Perijinan..... 38

3.3.2 Mengambil Data di Lapangan 38

3.4 Penyelesaian 39

3.4.1 Perencanaan..... 39

3.4.2 Pengolahan Data..... 39

3.4.3 Analisis Data 39

3.4.4 Pemilihan Program 41

3.5 Tahap Pelaporan Hasil..... 41

3.6 Tahap Penarikan Kesimpulan..... 41

BAB IV HASIL DAN PEMBAHASAN 42

4.1 Hasil Penelitian Algoritma Greedy dan Algoritma A*..... 43

4.1.1 Hasil Penelitian dengan Menggunakan Algoritma A* untuk Mengetahui Rute Terpendek 43

4.1.1.1 Rute Terpendek Algoritma A* Ditinjau Terhadap Jarak antar Daerah 43

4.1.1.1.1 Simpul-Simpul Pendistribusian..... 44

4.1.1.1.2 Jarak Euclidian untuk Mengetahui Nilai Suatu Simpul n ke Simpul Tujuan dengan Melihat *Latitude* dan *Longtitudenya*..... 44

UNNES
UNIVERSITAS NEGERI SEMARANG

4.1.1.1.3 Bobot Sisi untuk Mengetahui Nilai Setiap Sisinya 46
4.1.1.2 Rute Terpendek Algoritma A* Ditinjau Terhadap Kepadatan antar Daerah 60
4.1.2 Hasil Penelitian dengan Menggunakan Algoritma Greedy untuk Mengetahui Rute Terpendek 73
4.1.2.1 Rute Terpendek Algoritma Greedy Ditinjau Terhadap Jarak antar Daerah 73
4.1.2.2 Rute Terpendek Algoritma Greedy Ditinjau Terhadap Kepadatan antar Daerah 78
4.2 Menentukan Rute Terpendek dengan Menggunakan *Javascript*..... 82
4.2.1 *Input* Data 82
4.2.2 Tampilan Awal Program *Javascript*..... 83
4.2.3 Hasil Perhitungan Rute Terpendek dengan Bobot Jarak Menggunakan Algoritma A*..... 84
4.2.4 Hasil Perhitungan Rute Terpendek dengan Bobot Kepadatan Lalu Lintas Menggunakan Algoritma A* 85
4.2.5 Hasil Perhitungan Rute Tependek dengan Bobot Jarak Menggunakan Algoritma Greedy..... 86
4.2.6 Hasil Perhitungan Rute Terpendek dengan Bobot Kepadatan Lalu Lintas Menggunakan Algoritma Greedy 87
BAB V PENUTUP..... 88
5.1 Simpulan..... 88

UNNES
UNIVERSITAS NEGERI SEMARANG

5.2 Saran 90

DAFTAR PUSTAKA 92

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR TABEL

Tabel 2.1 Ekivalensi Mobil Penumpang 29

Tabel 2.2 Elemen pada *Javascript Grammar* 33

Tabel 2.3 Properti pada Jendela Web *Browser* Oleh *Javascript* 34

Tabel 4.1 Simpul pada Pendistribusian Air Minum Isi Ulang *Home Industry* Rizqua 44

Tabel 4.2 Jarak *Euclidian* pada Pendistribusian Air Minum Isi Ulang *Home Industry* Rizqua 45

Tabel 4.3 Jarak pada Pendistribusian Air Minum Isi Ulang *Home Industry* Rizqua 46

Tabel 4.4 Kepadatan Lalu Lintas pada Pendistribusian Air Minum Isi Ulang *Home Industry* Rizqua 60

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR GAMBAR

Gambar 2.1 Jalan pada Graf G ($v_3 \rightarrow v_5 \rightarrow v_4 \rightarrow v_2 \rightarrow v_1$) 14

Gambar 2.2 Graf G 16

Gambar 2.3 Graf Sederhana 17

Gambar 2.4 Graf Berarah 18

Gambar 2.5 Graf Tidak Berarah 18

Gambar 2.6 Graf Lengkap dengan 5 Simpul 19

Gambar 2.7 Graf Kosong dengan 4 Simpul 19

Gambar 2.8 Graf G Bipartisi, Graf $K_{3,2}$ Bipartisi Komplit 20

Gambar 2.9 Eulerian pada Graf 21

Gambar 2.10 Graf Berlabel 22

Gambar 2.11 V_1 Bertetanggaan dengan V_2 dan V_4 22

Gambar 2.12 Sisi Ganda 23

Gambar 3.1 Diagram Alur Kerja Penelitian 36

Gambar 3.2 Diagram Alur Algoritma A* 40

Gambar 4.1 Rute Awal Pendistribusian Air Minum Isi Ulang dari *Home Industry* Rizqua Kabupaten Pekalongan 43

Gambar 4.2 Rute Awal dengan Bobot Jarak Pendistribusian Air Minum Isi Ulang di *Home Industry* Kabupaten Pekalongan 46

Gambar 4.3 Rute Terpendek Algoritma A* dengan Bobot Jarak Pendistribusian Air Minum Isi Ulang di *Home Industry* Kabupaten Pekalongan 59

Gambar 4.4 Rute Awal dengan Bobot Kepadatan Lalu Lintas

Pendistribusian Air Minum Isi Ulang di *Home Industry* Kabupaten Pekalongan 61

Gambar 4.5 Rute Terpendek Algoritma A* dengan Bobot Kepadatan Lalu Lintas Pendistribusian Air Minum Isi Ulang di *Home Industry* Kabupaten Pekalongan 73

Gambar 4.6 Rute Terpendek Algoritma Greedy dengan Bobot Jarak Pendistribusian Air Minum Isi Ulang di *Home Industry* Kabupaten Pekalongan 77

Gambar 4.7 Rute Terpendek Algoritma Greedy dengan Bobot Kepadatan Lalu Lintas Pendistribusian Air Minum Isi Ulang di *Home Industry* Kabupaten Pekalongan 82

Gambar 4.8 Tampilan *Input* Data di *Notepad* 82

Gambar 4.9 Tampilan Awal Program Algoritma A* 83

Gambar 4.10 Tampilan Awal Program Algoritma Greedy 83

Gambar 4.11 Hasil Perhitungan Rute Terpendek dengan Bobot Jarak Menggunakan Algoritma A* 84

Gambar 4.12 Hasil Perhitungan Rute Terpendek dengan Bobot Kepadatan Lalu Lintas Menggunakan Algoritma A* 85

Gambar 4.13 Hasil Perhitungan Rute Terpendek dengan Bobot Jarak Menggunakan Algoritma Greedy 86

Gambar 4.14 Hasil Perhitungan Rute Terpendek dengan Bobot Kepadatan Lalu Lintas Menggunakan Algoritma Greedy 87

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR LAMPIRAN

Lampiran 1 *Source Code Javascript* Algoritma A* 94

Lampiran 2 *Source Code Javascript* Algoritma Greedy 109

Lampiran 3 *Source Code Logic Javascript* 114

Lampiran 4 Data Jarak dan Kepadatan Lalu Lintas 127

Lampiran 5 Rute Penditribusian Air Minum Isi Ulang *Home Industry* Rizqua di Kabupaten Pekalongan dan Jalan Alternatif 128

Lampiran 6 Rute Penditribusian Air Minum Isi Ulang *Home Industry* Rizqua di Kabupaten Pekalongan dengan Menggunakan Algoritma Greedy dan Algoritma A* 129

UNNES
UNIVERSITAS NEGERI SEMARANG

UNNES
UNIVERSITAS NEGERI SEMARANG

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Matematika adalah cabang ilmu pengetahuan yang berperan penting dalam perkembangan dunia. Pentingnya matematika tidak lepas dari perannya dalam segala jenis dimensi kehidupan. Matematika juga seringkali dibutuhkan untuk menunjang eksistensi ilmu-ilmu lain seperti fisika, kimia, astronomi, biologi, ekonomi, dan lain sebagainya.

Teori graf merupakan cabang dari matematika yang sebenarnya sudah ada sejak lebih dari dua ratus tahun silam. Jurnal pertama tentang teori graf muncul pada tahun 1936, oleh matematikaawan terkenal dari Swiss bernama Euler. (Cahyaningsih *et al*., 2013: 121-126).

Menemukan rute terpendek adalah usaha untuk mencari rute yang paling pendek jaraknya dari posisi awal hingga akhir dengan nilai yang paling kecil dibandingkan dengan seluruh rute yang ada. Dalam penelitian ini saya mengulas tentang sikel hamilton tentunya dengan rute yang paling pendek diantara rute-rute yang lain. Persoalan sikel terpendek antara beberapa titik (simpul) dan garis (sisi) yang berhubungan dengan dimulai dari simpul awal dan kembali ke simpul awal lagi. Mencari sikel terpendek di dalam graf merupakan salah satu persoalan optimasi. Persoalan ini bisa direpresentasikan dalam bentuk graf. Graf yang digunakan dalam pencarian keoptimalan sikel ini adalah graf berbobot, yaitu graf

yang setiap sisinya diberikan suatu nilai. Nilai pada sisi graf dapat menyatakan suatu jarak antar simpul.

Asumsi yang saya gunakan di sini adalah bahwa semua nilai bernilai positif, kata terpendek di sini diartikan meminimalisasi nilai pada suatu rute di dalam graf. Secara umum, pencarian rute terpendek dapat dibagi menjadi dua metode yaitu metode konvensional dan metode heuristik. Metode konvensional cenderung lebih mudah dipahami daripada metode heuristik, tetapi jika dibandingkan, hasil yang diperoleh dari metode heuristik lebih variatif dan dengan waktu perhitungan yang lebih singkat (Kreshna, 2011). Secara konsep algoritma, metode konvensional lebih mudah untuk dipahami tetapi hasil yang diperoleh metode heuristik lebih variatif. Dengan metode heuristik, waktu perhitungan yang diperlukan lebih cepat 30% dibandingkan dengan metode konvensional (Mutakhiroh *et al*., 2007).

Metode heuristik adalah teknik yang dirancang untuk menyelesaikan masalah yang mengabaikan apakah solusi dapat dibuktikan dengan benar, tapi menghasilkan solusi yang baik atau memecahkan masalah yang lebih sederhana yang mengandung atau memotong dengan pemecahan masalah yang lebih kompleks. Metode heuristik bertujuan untuk mendapatkan performa komputasi yang berpotensi pada keakuratan dan keminimailisasian biaya.

Best-first search merupakan sebuah metode yang membangkitkan simpul dari simpul sebelumnya. *Best-first search* memilih simpul baru yang memiliki biaya terkecil diantara semua *leaf simpuls* (simpul-simpul pada level terdalam) yang pernah dibangkitkan. Penentuan simpul terbaik dilakukan dengan

menggunakan sebuah fungsi yang disebut fungsi evaluasi *f(n)* (Russel & Norvig, 1995). Metode ini merupakan kombinasi dari metode *depth-first search* dan *breadth-first search*. Proses dilakukan dengan melakukan penelusuran terhadap setiap simpul (*simpul)* yang memiliki perkiraan rute terpendek. Pada metode *best-first search*, pencarian diperbolehkan mengunjungi *simpul* yang ada di level yang lebih rendah, jika ternyata *simpul* pada level yang lebih tinggi ternyata memiliki nilai heuristik yang lebih buruk. Pada proses *searching* ini dilakukan dengan cara memberikan perkiraan berapa jauh *simpul* asal dari solusi yang diinginkan.

Pada *best-first search* jika suksesor digunakan, maka dikatakan algoritma mengembangkan simpul tersebut. Setiap sebuah simpul dikembangkan, algoritma akan menyimpan setiap suksesor simpul *n* sekaligus dengan harga (*cost*) dan petunjuk pendahuluanya yang disebut *parent*. Algoritma akan berakhir pada simpul tujuan dan tidak ada lagi pengembangan simpul. Fungsi evaluasi pada *best-first search* dapat berupa informasi biaya perkiraan dari suatu simpul menuju simpul tujuan atau gabungan antara biaya sebenarnya dan biaya perkiraan tersebut. Biaya perkiraan dapat diperoleh dengan menggunakan suatu fungsi yang disebut fungsi heuristik. Pada strategi *best-first search*, *cost* sebenarnya yaitu dari simpul awal ke simpul *n* dinotasikan dengan *g(n)* dan fungsi heuristik yang digunakan yaitu perkiraan nilai dari simpul *n* ke simpul tujuan dinotasikan dengan *h(n)*. Algoritma yang menggunakan metode *best-first search* yaitu algoritma Greedy dan algoritma A*.

Algoritma Greedy merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara

pada setiap langkahnya dengan memanfaatkan algoritma Greedy akan diketahui rute terpendek pendistribusian air minum isi ulang yang akan dipasarkan di Kabupaten Pekalongan sehingga dapat membantu *home industry* untuk menentukan rute terpendek agar pemasarannya lebih efisien. Algoritma A* adalah salah satu pencarian untuk memberikan solusi yang cukup bagus dan akurat untuk pemasalahan kehidupan sehari-hari, seperti permasalahan pemasaran. Algoritma A* merupakan jenis algoritma untuk menentukan rute. Berbeda dengan algoritma Greedy, algoritma A* menghitung semua kemungkinan dan menyimpannya sehingga jika setiap memilih jalan. Algoritma A* juga membandingkan dengan jalan lain yang disimpan. Sehingga hasil pencarian sikel terpendek dengan menggunakan algoritma ini akan menghasilkan hasil yang lebih optimum.

Home industry adalah rumah usaha produk barang atau juga perusahaan kecil. Dikatakan sebagai perusahaan kecil karena jenis kegiatan ekonomi ini dipusatkan di rumah. Pada umumnya pelaku kegiatan ekonomi yang berbasis di rumah ini adalah keluarga itu sendiri ataupun salah satu dari anggota keluarga yang berdomisili di tempat tinggalnya itu dengan mengajak beberapa orang di sekitarnya sebagai karyawannya. Meskipun dalam skala kecil, namun kegiatan ekonomi ini secara tidak langsung membuka lapangan pekerjaan untuk sanak saudara ataupun tetangga di kampung halamannya. Dengan begitu, *home industry* ini otomatis dapat membantu program pemerintahan dalam upaya mengurangi angka pengangguran.

Dalam suatu perusahaan yang bergerak dalam bidang industri tentunya tidak lepas dari pemasaran produk yang biasa disebut pendistribusian produk. Pada

pendistribusian dari *home industry* ini tentunya akan memperkirakan biaya dan waktu. Semakin banyak tempat yang menjadi sasaran pemasaran akan semakin banyak pula biaya dan waktu yang akan didapat.

Oleh karena itu, diperlukan ilmu yang akan mempertimbangkan biaya dan waktu tersebut agar lebih efisien. Dalam masalah seperti ini diperlukan rute terpendek dari semua simpul pemasaran yang ada.

Pengambilan data adalah data sekunder yang diperoleh dari *home industry* Rizqua Kabupaten Pekalongan. Data yang diambil berupa lokasi pemasaran, selanjutnya dilakukan pecarian jarak dengan bantuan *Google Maps*. Analisis data dilakukan dengan menggunakan algoritma Greedy dan algoritma A*.

Berdasarkan latar belakang permasalah di atas, maka penulis mengambil judul **"PERBANDINGAN ALGORITMA GREEDY DAN ALGORITMA A* PADA PENYELESAIAN *TRAVELLING SALESMAN PROBLEM* (Studi Kasus: Pendistribusian Air Minum Isi Ulang di *Home Industry* Rizqua Kabupaten Pekalongan).**

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah dapat dirumuskan permasalahan yang akan diselesaikan dalam penelitian ini yaitu:

(1) Bagaimana penerapan algoritma Greedy dan algoritma A* untuk mencari rute terpendek pendistribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan?

UNNES
UNIVERSITAS NEGERI SEMARANG

(2) Apakah algoritma Greedy dan algoritma A* dapat menyelesaikan masalah pendistribusian air minum isi ulang *home industry* Rizqua di Kabupaten Pekalongan?

(3) Bagaimana perbandingan algoritma Greedy dan algoritma A* pada permasalahan pendistribusian pada *Travelling Salesman Problem*?

1.3 Batasan Masalah

(1) *Travelling Salesman Problem* (TSP) diterapkan hanya untuk pencarian rute terpendek dari beberapa objek pemasaran air minum isi ulang yang telah ditentukan Rizqua di Kabupaten Pekalongan.

(2) Koordinat tempat diperoleh dari aplikasi *Google Maps*.

(3) Jalan yang digunakan adalah jalan yang telah tercantum dalam *Google Maps*.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

(1) Mengetahui rute terpendek pendistribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan menggunakan algoritma Greedy dan algoritma A*.

(2) Menganalisis algoritma Greedy dan algoritma A* dalam penyelesaian masalah penditribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan.

(3) Mengetahui algoritma mana yang lebih efisien pada permasalahan pendistribusian pada *Travelling Salesman Problem.*

1.5 Manfaat Penelitian

(1) Bagi Penulis

Mengetahui dan memahami aplikasi teori graf terutama algoritma Greedy dan algoritma A* terhadap jaringan jalan di Kabupaten Pekalongan.

(2) Bagi Universitas

Dari hasil penelitian ini dapat menjadi referensi yang berkaitan dengan teori graf dalam menyelesaikan masalah menghitung sikel hamilton terpendek.

(3) Bagi *Home Industry* Rizqua

Memberikan informasi jalan yang harus dilewati agar sampai ke tempat tujuan dengan jarak dan biaya yang efisiensi serta dapat menghemat waktu perjalanan.

1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari beberapa bagian utama sebagai berikut:

BAB I: PENDAHULUAN

Bab ini membahas latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Bab ini membahas mengenai tinjauan pustaka yang berfungsi sebagai sumber dalam memahami permasalahan yang berkaitan dengan teori graf, *Travelling Salesman Problem*, algoritma Greedy, dan algoritma A*.

BAB III: METODE PENELITIAN

Bab ini merupakan prosedur dan langkah-langkah yang harus dilakukan sebelum penelitian sehingga dapat terarah dan terlaksana dengan baik dalam hal pelapor penelitan.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini merupakan pembahasan dari hasil penelitian yang berupa penerapan algoritma Greedy dan algoritma A* untuk optimasi rute pendistribusian di Kabupaten Pekalongan dengan perhitungan secara manual.

BAB V: PENUTUP

Bab ini membahas mengenai kesimpulan dan saran yang berupa hasil dari penelitian rute terpendek menggunakan algoritma Greedy dan algoritma A* pada penyelesaian *Travelling Salesman Problem* untuk menyelesaikan pendistribusian air minum isi ulang *home industry* Rizqua Kabupaten Pekalongan.

UNNES
UNIVERSITAS NEGERI SEMARANG

BAB II

TINJAUAN PUSTAKA

2.1 Program *Linear*

Program *linear* merupakan peralatan standar yang telah menghemat ribuan atau jutaan dolar bagi banyak perusahaan bahkan bagi perusahaan yang sedang besarnya di berbagai negara industri dan pemakainya di sektor-sektor lain masyarakat meluas dengan cepat. Pemrograman *linear* memakai suatu model matematis untuk menggambarkan masalah yang dihadapi kata sifat *"linear"* berarti bahwa semua fungsi matematis dalam model ini harus merupakan fungsi-fungsi *linear*.

Kata "program" di sini merupakan sinonim untuk kata "perencanaan". Maka membuat pemrogram *linear* adalah membuat rencana kegiatan-kegiatan untuk memperoleh hasil yang optimal, ialah suatu hasil yang mencapai tujuan yang ditentukan dengan cara yang paling baik (sesuai model matematis) di antara semua alternatif yang mungkin. Jadi, pengertian program *linear* adalah suatu teknik perencanaan yang bersifat analisis yang analisisnya menggunakan model matematis dengan tujuan menemukan beberapa kombinasi alternatif pemecah optimum terhadap persoalan.

2.2 Riset Operasi

Istilah Riset Operasi pertama kali digunakan pada tahun 1940 oleh Mc Closky dan Trefthen di suatu kota kecil, Bowdsey, Inggris. Pada masa awal perang 1939, pemimpin militer Inggris memanggil sekelompok ahli-ahli sipil dari

berbagai disiplin dan megkoordinasikan mereka ke dalam suatu kelompok yang diserahi tugas mencari cara-cara yang efisien untuk menggunakan alat yang baru ditemukan yang dinamakan radar dalam suatu sistem peringatan dini menghadapi serangan udara. Kelompok ahli Inggris ini dan kelompok-kelompok lain berikutnya melakukan penelitian (*research*) pada operasi-operasi (*operations*) militer. Hasilnya sangat memuaskan, kesuksesan proyek manajemen radar ini menyebabkan pemimpin militer lebih mengandalkan riset operasi dalam membuat suatu keputusan operasional yang penting (Hilier & Lieberman, 1990: 4).

Setelah perang, keberhasilan kelompok-kelompok penelitian operasi-operasi di bidang militer menarik perhatian para industriawan yang sedang mencari penyelesaian terhadap masalah-masalah yang rumit. Pada tahun lima puluhan baik di Inggris maupun Amerika Serikat, adalah suatu dasa warsa penting dalam sejarah Riset Operasi. Selama periode ini, teknik-teknik program linear dan dinamik telah ditemukan dan diperluas. Langkah besar terjadi dalam penelitian murni tentang masalah persediaan produksi dan antri (*queueing*) (Mulyono, 2004: 1-2).

Riset operasi merupakan pengambilan keputusan dengan memanfaatkan pengetahuan ilmiah melalui usaha kelompok antar disiplin yang bertujuan untuk menentukan peggunaan terbaik sumber daya yang terbatas. Model riset operasi berkaitan dengan data deterministik biasanya jauh lebih sederhana dari pada yang melibatkan data probabilistik (Taha, 1997: 4).

Menurut Morse dan Kimball, *Method Of Operation* Research, riset operasi adalah metode ilmiah (*scientific* method) yang memungkinkan para manajer

UNNES
UNIVERSITAS NEGERI SEMARANG

mengambil keputusan mengenai kegiatan yang mereka tangani dengan dasar kuantitatif. Sedangkan menurut Miller dan M.K.Star, *Executive Decisions and Operation Research,* riset operasi adalah berkaitan dengan menentukan pilihan secara ilmiah bagaimana merancang dan menjalankan sistem manusia mesin secara terbalik, biasanya membutukan alokasi sumber daya yang langka (So *et al*., 2013: 812-820).

Secara umum dapat diartikan bahwa riset operasi berkaitan dengan proses pengambilan keputusan yang optimal dalam penyusunan model dari sistem-sistem baik *determinisik* maupun *probabilistik*. Sebagaimana ditunjukkan oleh namanya riset operasi meliputi "riset mengenai operasi". Nama ini menyatakan sesuatu mengenai pendekatan dan bidang aplikasi dari bidang ini, maka riset operasi diterapkan kepada masalah-masalah mengenai bagaimana melaksanakan dan mengkoordinasi operasi atau kegiatan-kegiatan dalam suatu organisasi. Sifat dari organisasi pada dasarnya adalah tidak material dan sebenarnya riset operasi secara luas dipakai dalam bisnis industri, ketentaraan pemerintahan sipil dan lembaga-lembaga, rumah sakit, dan sebagainya.

2.3 *Travelling Salesman Problem* (*TSP*)

Travelling Salesman Problem (TSP) termasuk ke dalam persoalan yang sangat terkenal dalam teori graf. Nama persoalan ini diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota (Wicaksana *et al*., 2014). Uraian persoalannya adalah sebagai berikut, pedagang menggunakan waktunya untuk mengunjungi *n* kota (simpul) secara siklus perputaran. Kebanyakan *Travelling Salesman Problem* merupakan suatu simetris yang berarti

untuk dua kota A dan B jarak dari kota A ke kota B adalah sama dengan jarak dari kota B ke kota A dalam hal ini kita akan mendapatkan panjang perjalanan keliling yang sama persis jika kita membalikan sikel perjalanan tersebut.

Kode program komputer yang dibuat untuk menyelesaikan persoalan TSP telah berkembang semakin baik dari tahun ke tahun. Tanda yang paling mencolok dari perkembangan metode untuk menyelesaikan persoalan TSP adalah bertambahnya jumlah simpul (*simpul*) yang dapat diselesaikan mulai dari solusi persoalan 49 kota yang dikembangkan oleh Dantzig Fulkerson, dan Johnson's pada tahun 1954 sampai kepada solusi persoalan 24.978 kota pada tahun 2004. data-data tersebut didapat dari *National Imagery and Mapping Agency* sebuah *database* nasional yang menyimpan nama-nama fitur geografi (Zulfikar, 2008: 5).

2.3.1 Konsep *Travelling Salesman Problem*

Apa yang dilakukan dalam TSP adalah membentuk sebuah *tour*. Operator yang bisa digunakan untuk masalah TSP adalah pencarian urutan semua lokasi untuk memilih lokasi yang belum pernah terpilih satu demi satu sehingga dihasilkan satu rute kunjungan yang lengkap dari lokasi awal kemudian mengunjungi semua lokasi yang lain tepat satu kali dan akhirnya kembali ke lokasi awal (Wiyanti, 2013: 1-6). Sehingga dengan definisi tersebut dapat dikatakan bahwa konsep permasalahan TSP memiliki aturan sebagai berikut:

1. Harus mengunjungi setiap kota tepat satu kali, tidak boleh kurang ataupun lebih.
2. Semua kota harus dikunjungi dalam satu kali perjalanan (*tour*).
3. Dimulai dan diakhiri pada kota yang sama.

UNNES
UNIVERSITAS NEGERI SEMARANG

2.4 Rute Terpendek

Setiap dua titik terjadi beberapa jalan atau pun lintasan pada graf, dimana jalan atau pun lintasan dengan bobot yang minimum disebut sebagai Rute Terpendek. Bobot di sini dapat diartikan jarak, waktu tempuh atau ongkos transportasi dari satu titik ke titik lainnya yang berbentuk rute tertentu (Dimyati, 1990:164).

2.5 Teori Graf

Graf merupakan suatu cabang kajian ilmu yang memiliki banyak terapan yang mempunyai sifat-sifat graf atau grafik. Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul atau simpul atau *vertex* atau *simpul* yang terhubung oleh sisi atau busur atau *edge* atau *arc*. Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan "simpul") yang dihubungkan oleh garis-garis (melambangkan "sisi") atau garis berpanah (melambangkan "busur").

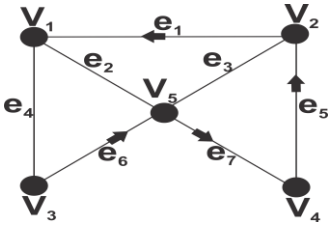
Suatu sisi dapat menghubungkan suatu simpul dengan simpul yang sama, sisi yang demikian disebut gelung atau *loop*. Sering juga graf digunakan untuk merepresentasikan suatu jaringan. Misalkan jaringan jalan raya dimodelkan graf dengan kota sebagai simpul yang nilainya adalah panjang dari jalan tersebut. Misalkan simpul merepresentasikan kota, sisi merepresentasikan perjalanan yang memungkinkan, dan nilai dari setiap sisi adalah jarak yang ditempuh dalam perjalanan tersebut. Graf G didefinisikan sebagai pasangan himpunan (V, E) ditulis dengan notasi G = (V, E), yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul dan E adalah himpunan sisi yang menghubungkan sepasang simpul. V tidak boleh kosong, sedangkan E boleh kosong.

Simpul pada graf dapat dinomori dengan huruf seperti a, b, c, ..., v, w, ..., dengan bilangan asli 1, 2, 3, ... atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau (v, u) atau [u, v] atau [v, u] atau u, v atau v, u atau dinyatakan dengan lambang e_1, e_2, ... Dengan kata lain, jika e adalah sisi yang menghubungkan simpul u dengan v, maka e dapat ditulis sebagai e = (u, v). Secara geometri graf digambarkan sebagai sekumpulan simpul di dalam bidang dwimitra yang dihubungkan dengan sekumpulan sisi.

2.5.1 Keterhubungan pada Graf

2.5.1.1 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, e_3, v_3, ..., e_k, v_k) yang suku-sukunya bergantian sedemikian hingga v_{i-1} dan v_i adalah simpul-simpul akhir sisi e_i untuk $1 \leq i \leq k$. Dikatakan W adalah sebuah jalan dari simpul v_0 kesimpul v_k, atau jalan (v_0, v_k). Simpul v_0 dan simpul v_k berturut-turut disebut simpul awal dan simpul akhir W. Sedangkan simpul-simpul v_1, v_2, v_3, ..., v_{k-1} disebut simpul 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 simpul awal dan simpul akhir dari W identik (sama) (Budayasa, 2007: 6).



Gambar 2.1 Jalan pada Graf G ($v_3 \rightarrow v_5 \rightarrow v_4 \rightarrow v_2 \rightarrow v_1$)

UNNES
UNIVERSITAS NEGERI SEMARANG

2.5.1.2 Jejak (Trail)

Sebuah simpul G, mungkin saja muncul lebih dari satu kali dalam jalan W, begitu juga dengan sebuah sisi G, boleh muncul lebih dari satu kali pada jalan W. Jika semua sisi e_1, e_2, e_3, ..., e_k dalam jalan W berbeda, maka W disebut sebuah jejak (*Trail*) (Budayasa, 2007: 6). Perhatikan Gambar 2.1, jejak pada graf G ($v_3 \rightarrow v_5 \rightarrow v_2 \rightarrow v_1 \rightarrow v_5 \rightarrow v_4$).

2.5.1.3 Lintasan (Path)

Jika semua simpul v_0, v_1, v_2, ..., v_k dalam jejak W berbeda, maka W disebut lintasan (*path*) (Budayasa, 2007: 6). Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G ialah barisan berselang-seling, simpul-simpul dan sisi-sisi yang terbentuk $v_0, e_1, v_1, e_2, v_2, ..., v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ..., $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G. Jika graf yang ditinjau adalah graf sedehana, maka kita cukup menuliskan lintasan sebagai barisan simpul-simpul saja $v_0, e_1, v_1, e_2, v_2, ..., v_{n-1}, e_n, v_n$ karena antara dua buah simpul berurutan di dalam lintasan tersebut hanya ada satu sisi. Simpul dan sisi yang dilalui di dalam lintasan tersebut hanya ada satu sisi. Simpul dan sisi yang dilalui di dalam lintasan tidak boleh berulang. Sebuah lintasan dikatakan lintasan sederhana (*simple path*) jika semua simpulnya berbeda (setiap sisi yang dilalui hanya satu kali). Panjang lintasan adalah jumlah sisi dalam lintasan tersebut. Perhatikan Gambar 2.1, lintasan pada graf G ($v_3 \rightarrow v_5 \rightarrow v_4 \rightarrow v_2 \rightarrow v_1$).

2.5.1.4 Sikel

Sebuah sikel (*cycle*) adalah sebuah jejak tertutup (*closed trail*) yang titik awal dan semua titik internalnya berbeda. Banyaknya sisi dalam suatu sikel

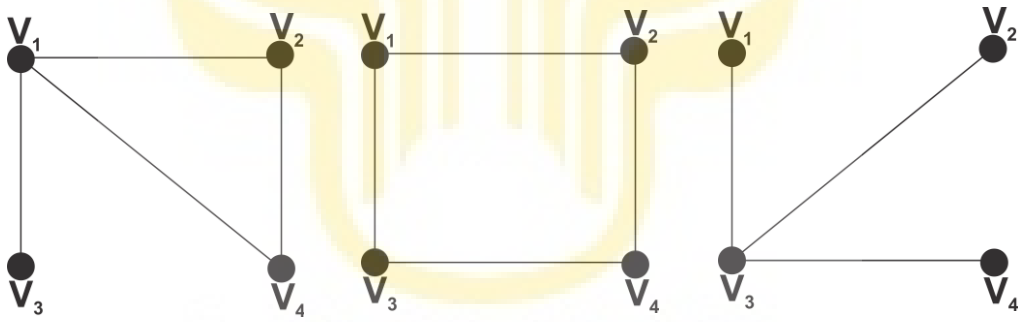
UNNES
UNIVERSITAS NEGERI SEMARANG

disebut panjang dari sikel tersebut. Sikel dengan panjang k disebut sikel-k, disimbolkan dengan C_k (Budayasa, 2007: 6).

Di dalam penelitian ini sikel disebut juga rute. Perhatikan Gambar 2.1, sikel pada graf G ($v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_4 \rightarrow v_2 \rightarrow v_1$).

2.5.1.5 Lintasan Hamilton dan Sikel Hamilton

Misalkan G sebuah lintasan di G yang memuat semua simpul di G disebut lintasan hamilton. Graf non hamilton yang memuat lintasan hamilton disebut graf semi-hamilton. Sikel hamilton adalah sikel yang melalui tiap simpul di graf tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali. Graf hamilton adalah graf yang memiliki sikel hamilton (Munir, 2001: 232).



(a) (b) (c)

Gambar 2.2 Graf G

Keterangan:

(a) Graf G memiliki lintasan hamilton $v_4 \rightarrow v_2 \rightarrow v_1 \rightarrow v_3$.

(b) Graf G memiliki sikel hamilton $v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_3 \rightarrow v_1$.

(c) Graf G tidak lintasan hamilton maupun sikel hamilton.

Persoalan mencari sikel terpendek di dalam graf merupakan salah satu persoalan optimasi. Graf yang digunakan dalam pencarian sikel terpendek adalah

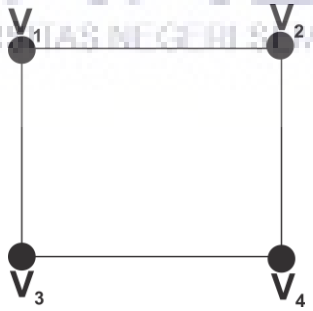
graf bernilai, yaitu graf yang setiap sisinya diberikan suatu nilai. Nilai pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan dan sebagainya. Asumsi yang digunakan di sini adalah bahwa semua nilai bernilai positif. Sikel terpendek adalah jalur yang dilalui dari suatu *simpul* ke *simpul* lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari *simpul* awal ke *simpul* akhir paling kecil. Sikel terpendek adalah sikel minimum yang diperlukan untuk menempuh tempat dari satu simpul dan kembali kesimpul semula. Sikel minimum yang dimaksud dapat dicari dengan menggunakan graf.

2.5.2 Jenis-Jenis Graf

2.5.2.1 Berdasarkan Ada Tidaknya Gelung atau Sisi Ganda

2.5.2.1.1 Graf Sederhana (*Simple Graph*)

Graf yang tidak mengandung gelung dan sisi ganda dinamakan graf sederhana (*simple graph*). Pada graf ini sisi merupakan pasangan tak terurut (*unordered pairs*) sehingga jika menuliskan sisi (u, v) sama saja dengan (v, u) dan G = (V, E) terdiri dari himpunan tidak kosong simpul–simpul dan E adalah himpunan pasangan tak terurut yang berbeda yang disebut sisi.



Gambar 2.3 Graf Sederhana

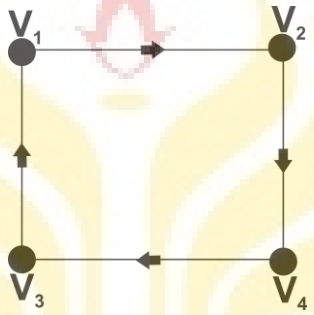
2.5.2.1.2 Graf Tidak Sederhana (*Unsimple Graph*)

Graf yang mengandung sisi ganda atau gelung dinamakan graf tak sederhana (*unsimple graph* atau *multigraph*).

2.5.2.2 Berdasarkan Orientasi Arah pada Sisi

2.5.2.2.1 Graf Berarah (*Directed Graph* atau *Digraph*)

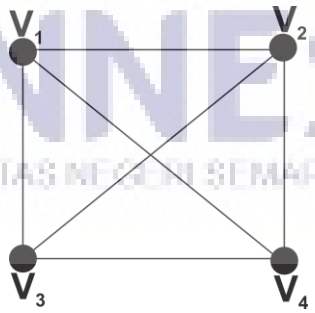
Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.



Gambar 2.4 Graf Berarah

2.5.2.2.2 Graf Tidak Berarah (*Undirected Graph*)

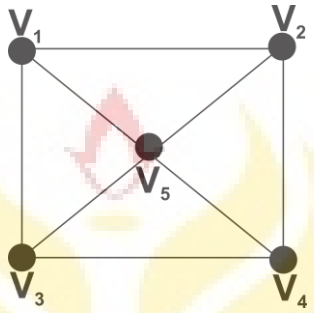
Graf yang sisinya tidak mempunyai orientasi arah disebut sebagai graf tak berarah.



Gambar 2.5 Graf Tidak Berarah

2.5.2.3 Graf Lengkap (Graf Komplit)

Graf lengkap ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Sebuah graf lengkap dengan *n* buah simpul dilambangkan dengan K_n. Setiap simpul pada K_n berderajat n-1, sehingga jumlah sisi yang ada adalah $\frac{n(n+1)}{2}$.



Gambar 2.6 Graf Lengkap dengan 5 Simpul

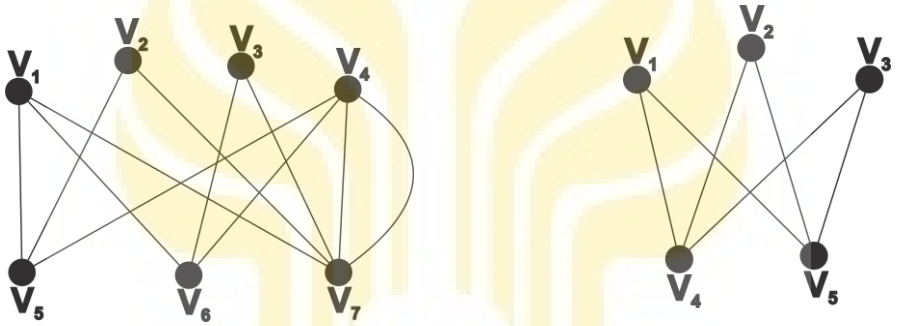
2.5.2.4 Graf Kosong

Graf yang tidak memiliki sisi disebut graf kosong atau graf nol. Graf nol dengan *n* simpul dilambangkan dengan N_n. Misalnya, graf kosong dengan 4 simpul. Diperlihatkan pada gambar 2.7 dibawah.

Gambar 2.7 Graf Kosong dengan 4 Simpul

2.5.2.5 Graf Bipartisi

Sebuah Graf G disebut graf bipartisi jika himpunan simpul G dapat dipartisi menjadi dua himpunan bagian A dan B sedemikian hingga setiap sisi dari G menghubungkan sebuah simpul di A dan sebuah simpul di B. Kita sebut (A, B) bipartisi dari G selanjutnya apabila G sedehana dan bipartisi dengan bipartisi (A, B) sedemikian hingga setiap simpul di A berhubungan langsung dengan setiap simpul di B, maka G disebut graf bipartisi komplit, dilambangkan dengan $K_{m,n}$ di mana |A| = m dan |B| = n. Sebagai contoh perhatikan graf pada gambar:



G $K_{3,2}$

Gambar 2.8 Graf G Bipartisi. Graph $K_{3,2}$ Bipartisi Komplit.

2.5.2.6 Graf Terhubung (Connected Graph) dan Graf Tak Terhubung (Disconnected Graph)

Suatu graf disebut sebagai graf terhubung apabila untuk setiap pasang simpul u dan v didalam himpunan V terdapat lintasan dari u ke v yang juga harus berarti ada lintasan dari v ke u. Jika tidak, maka G disebut graf tak terhubung (*disconnected graph*).

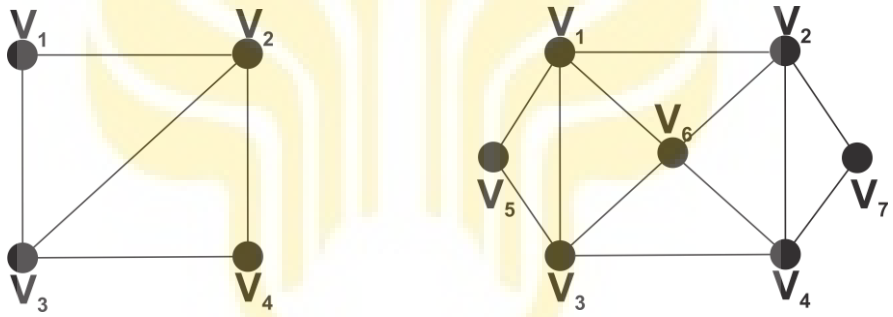
2.5.2.7 Sub Graf

Subgraf (*Upgraf*) merupakan sebuah graf yang ada pada sebuah graf yang lain. Misalkan bilamana sebuah graf G = (V, E), maka $G_1 = (V_1, E_1)$ merupakan subgraf dari G jika V_1, V dan E_1, E.

2.5.2.8 Graf Euler dan Graf Semi-Euler

Sebuah sikel di graf G yang memuat semua sisi G disebut sikel Euler. Jika graf G membuat semua sikel Euler, maka graf G disebut graf Euler.

Sebuah jejak buka yang memuat semua sisi graf disebut jejak Euler. Graf G disebut graf semi-Euler jika G memuat jejak Euler (Budayasa, 2007: 113).



(a) (b)

Gambar 2.9 Eulerian pada Graf

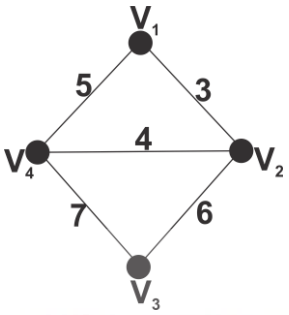
Keterangan:

(a) Graf G memiliki jejak Euler, yaitu $v_3 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_2$

(b) Graf G memiliki sikel Euler, yaitu $v_1 \rightarrow v_2 \rightarrow v_7 \rightarrow v_4 \rightarrow v_2 \rightarrow v_6 \rightarrow v_4 \rightarrow v_3 \rightarrow v_6 \rightarrow v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_1$.

UNNES
UNIVERSITAS NEGERI SEMARANG

2.5.2.9 Graf Berlabel



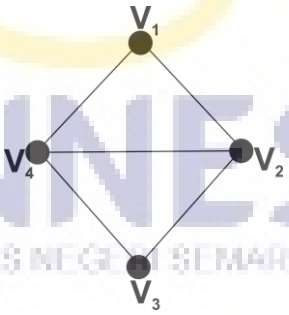
Gambar 2.10 Graf Berlabel

Graf berlabel atau berbobot adalah graf yang setiap sisinya mempunyai nilai atau bobot berupa bilangan positif.

2.5.3 Teminologi Graf

2.5.3.1 Ketetanggan

Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung. Pada gambar 2.11, simpul 1 bertetanggaan dengan simpul 2&4. Simpul 1 tidak bertetangga dengan simpul 3.



Gambar 2.11 V_1 Bertetanggan dengan V_2 dan V_4

2.5.3.2 Derajat (Degree)

Derajat suatu simpul adalah banyaknya sisi yang menghubungkan suatu simpul. Sedangkan derajat graf G adalah jumlah derajat semua simpul graf G. Lihat pada Gambar 2.11.

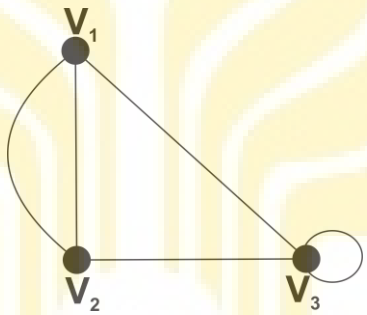
Keterangan:

$d(V_1) = d(V_3) = 2$

$d(V_2) = d(V_4) = 3$

2.5.3.3 Bersisian

Untuk sembarang ruas e = (v_j, v_k) dikatakan e bersisian dengan simpul v_j, atau e bersisian dengan simpul v_k. Pada Graf G_1, ruas (V_2, V_3) bersisian dengan simpul V_2 dan simpul V_3. Ruas (V_2, V_4) bersisian dengan simpul V_2 dan simpul V_4. Tetapi ruas (V_1, V_2) tidak bersisian dengan simpul V_4.



Gambar 2.12 Sisi Ganda

Keterangan:

$d(V_1) = 3$ bersisian dengan sisi ganda

$d(V_3) = 4$ bersisian dengan *self-loop* (derajat sebuah *self-loop* = 2)

UNNES
UNIVERSITAS NEGERI SEMARANG

2.6 Metode Heuristik

Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan masalah individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diingkan.

2.7 Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Langkah-langkah tersebut harus logis, ini berarti langkah-langkah tersebut harus dapat ditelusuri kebenarannya. Langkah-langkah yang tidak dapat memberikan solusi yang salah.

Donald E.Knuth, seorang penulis buku algoritma abad XX, sebagimana dikutipkan (Suarga, 2012:4), menyatakan ada beberapa ciri algoritma, sebagai berikut:

(1) Algoritma mempunyai awal dan akhir, suatu algoritma harus berhenti setelah mengerjakan serangkaian tugas. Dengan kata lain, suatu algoritma memiliki langkah terbatas.

(2) Setiap langkah pada algoritma harus didefinisikan dengan tepat sehingga tidak memiliki arti ganda (*not ambigious*),

(3) Algoritma Memiliki masukan (*input*) atau kondisi awal,

(4) Algoritma Memiliki keluaran (*output*) atau kondisi akhir, dan

(5) Algoritma harus efektif, bila diikuti benar-benar maka akan menyelesaikan persoalan.

UNNES
UNIVERSITAS NEGERI SEMARANG

2.8 Algoritma Greedy

Menurut Hayati (2014) algoritma Greedy adalah algoritma yang memecahkan masalah langkah demi langkah, pada setiap langkah:

(1) Mengambil pilihan yang terbaik yang dapat diperoleh saat itu.

(2) Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan mencapai optimum global. Algoritma Greedy mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global.

Prinsip algoritma Greedy adalah "*take what you can get now!*'. Ambil apa yang anda peroleh sekarang!. Prinsip ini juga dipakai dalam pemecahan masalah optimasi.

Persoalan optimasi dalam konteks algoritma Greedy disusun oleh elemen-elemen sebagai berikut:

(1) Himpunan Kandidat, C

Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

(2) Himpunan Solusi, S

Merupakan himpunan dari kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat.

(3) Fungsi Seleksi, dinyatakan sebagai predikat SELEKSI

Merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

(4) Fungsi Kelayakan (*feasible*), dinyatakan dengan predikat LAYAK

Merupakan fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama

UNNES
UNIVERSITAS NEGERI SEMARANG

dengan himpunan solusi yang sudah terbentuk tidak melanggar keadaan yang ada.

(5) Fungsi Objektif

Merupakan fungsi yang memaksimumkan atau meminumkan nilai solusi. Kita berharap optimum global merupakan solusi optimum dari persoalan. Namun, ada kalanya 2 optimum global belum tentu merupakan solusi optimum (terbaik), tetapi dapat merupakan solusi sub-optimum. Hal ini dapat dijelaskan dari dua faktor berikut:

(a) Algoritma Greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada.

(b) Pemilihan fungsi SELEKSI biasanya didasarkan pada fungsi objektif (fungsi SELEKSI bisa saja identik dengan fungsi objektif).

Jika fungsi SELEKSI tidak identik dengan fungsi objektif, kita harus memilih fungsi yang tepat untuk menghasilkan nilai optimum. Karena itu, pada sebagian masalah algoritma Greedy tidak selalu berhasil memberikan solusi yang benar-benar optimum. Tetapi, algoritma Greedy pasti memberikan solusi yang mendekati (*approximation*) nilai optimum.

UNNES
UNIVERSITAS NEGERI SEMARANG

Algoritma Greedy untuk mencari sikel terpendek dapat dirumuskan sebagai berikut:

a. Periksa semua sisi yang langsung bersisian dengan simpul A. Pilih sisi yang nilainya terkecil. Sisi ini menjadi sikel terpendek pertama, sebut saja L(1).

b. Tentukan lintasan terpendek kedua dengan cara berikut:

i. Hitung: d(i) = panjang L (1) + nilai sisi dari simpul akhir L(1) ke simpul ii yang lain

ii. Pilih d(i) yang terkecil

Bandingkan d(i) dengan nilai sisi (a,i).

Jika nilai sisi(a,i) lebih kecil daripada d(i), maka L(2)=L(1) U (sisi dari simpul akhir L(i) ke simpul i).

2.9 Algoritma A*

Algoritma A* adalah algoritma pencarian rute terpendek yang merupakan algoritma yang dituntun oleh fungsi heuristiknya, yang menentukan urutan simpul mana yang akan dikunjungi terlebih dahulu. Heuristik merupakan penilai yang memberi harga pada tiap simpul yang memandu A* mendapatkan solusi yang diinginkan.

Berbeda dengan algoritma Greedy, algoritma ini menghitung semua kemungkinan dan menyimpangnya sehingga jika setiap memilih jalan. Algoritma A* juga membandingkan dengan jalan lain yang disimpan. Sehingga hasil pencarian sikel terpendek dangan menggunakan algoritma ini akan menghasilkan hasil yang efisien. Namun karena ia terus membandingkan, algoritma ini memakan waktu yang cukup lama. Sehingga jika simpulnya sangat banyak akan memakan waktu yang sangat lama.

Menurut Supriyani (2008), berikut adalah langkah-langkah pencarian rute terpendek dengan algoritma A*:

(1) Dimulai dengan *open list* yang berisi simpul awal. Nilai *g* dari *simpul* tersebut menjadi 0 dan nilai *h* sesuai nilai heuristik. *Close list* berupa *list* kosong.

(2) Hingga *simpul* tujuan ditemukan, ulangi langkah berikut:

Jika tidak ada simpul pada *open list* maka laporkan kekeliruan. Jika tidak, ambil *simpul* pada *open list* yang memiliki nilai *f* terendah dan namakan sebagai *best simpul.* Pindahkan dari *open list*, masukan ke *close list*. Untuk setiap suksesor, lakukan langkah-langkah berikut ini:

a) Set suksesor menunjuk kembali ke *best simpul*. *Link* ke belakang ini memungkinkan jalur yang terbentuk dipulihkan setelah hasilnya ditemukan.

b) Hitung *g*(suksesor) = *g best simpul*+biaya yang diperlukan untuk mencapai hasil dari *best simpul* ke suksesor.

c) Jika suksesor berada di *open list* maka namakan sebagai *old* harus diubah apabila lintasan yang baru saja ditemukan menuju suksesor lebih baik dari lintasan yang ada sekarang.

d) Jika suksesor ada di *close list*, maka namakan sebagai *close list old* dan tambahkan ke suksesor dan *best simpul*. Periksa apakah lintasan yang baru ini lebih baik dari langkah sebelumnya, dan set *linkparent* bila *g* dan *f* tiap *simpul*, akhiri tiap cabang bila telah mencapai lintasan yang lebih baik dari sebelumnya. Tiap *link parent* dari *simpul* menujuk ke *parent* terbaik. Jika *simpul* nya menuju ke simpul awal lanjutkan penyebaran. Jika tidak maka nilai *g* nya mencerminkan lintasan yang lebih baik dan penyebaran dihentikan. Tetapi mungkin dengan nilai yang baru *g* yang *simpul* nya disebarkan ke bawah menjadi lintasan yang lebih baik dari lintasan sebelumnya.

UNNES
UNIVERSITAS NEGERI SEMARANG

e) Jika suksesor tidak pernah ada di *open list* atau *close list*, maka masukan ke *open list* dan tambahkan ke daftar suksesor dari *best simpul*.

2.10 Kapasitas Jalan

Kapasitas didefinisikan sebagai arus maksimum melalui suatu simpul di jalan yang dapat dipertahankan per satuan jam pada kondisi tertentu. Kapasitas lebih dikenal dengan "Daya tampung maksimal" suatu ruas jalan terhadap kapasitas volume lalu lintas yang melintas. Untuk jalan dua-jalur, dua-arah, kapasitas ditentukan untuk arus dua arah (kombinasi dua arah), tetapi untuk jalan dengan banyak jalur, arus dipisahkan per arah dan kapasitas ditentukan per jalur (MKJI, 1997:5-18).

2.11 Arus Lalu Lintas

Perhitungan dilakukan per satuan jam untuk satu atau lebih periode, misalkan didasarkan pada kondisi arus lalu-lintas rencana jam puncak pagi, siang, sore. Arus lalu-lintas untuk setiap gerakan dikonversi dari kendaraan per-jam menjadi satuan mobil penumpang (smp) per-jam dengan menggunakan ekivalensi mobil penumpang (emp) untuk masing-masing kendaraan (MKJI:1997).

Tabel 2.1 Ekivalensi Mobil Penumpang

Jenis Kendaraan	Ekivalensi Mobil Penumpang
Kendaraan Ringan	1,0
Kendaraan Berat	1,3
Sepeda Motor	0,4

2.12 Kecepatan

Kecepatan menggambarkan tingkat pergerakan kendaraan yang dinyatakan dalam jarak tempuh persatuan waktu atau nilai perubahan jarak terhadap waktu. Satuannnya adalah kilometer/jam, meter/detik.

2.13 Volume Jalan

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 dikaitkan sebagi 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 *Annual Average Daily Traffic Volume* (AADT).

Volume dipengaruhi oleh panjang pendeknya waktu perhitungan dan waktu perhitungan harus cukup panjang. Untuk menjaga agar variasi-variasi jangka pendek jangan sampai mempengaruhi rata-rata. Sehingga untuk mencari volume jalan adalah panjang rute dikalikan dengan ekivalensi mobil penumpang.

2.14 Kepadatan Jalan

Kepadatan diartikan sebagai jumlah kendaraan yang ada pada satu ruas jalan raya atau jalur biasanya dinyatakan sebagai jumlah kendaraan per kilometer atau satuan mobil penumpang per kilometer (smp/km).

Kepadatan sukar diukur secara langsung, karena diperlukan simpul ketinggian tertentu yang dapat mengamati jumlah kendaraan dalam panjang ruas jalan

tertentu, sehingga besarnya ditentukan dari dua parameter volume dan kecepatan yang mempunyai hubungan sebagi berikut.

Kepadatan menunjukan kemudahan bagi kendaraan untuk bergerak, seperti pindah jalur dan memilih kecepatan yang diinginkan.

Sehingga untuk mencari kepadatan jalan adalah membagi volume dengan kecepatan dan mengalikan dengan panjang rutenya.

2.15 *Javascript*

2.15.1 Mengenal *Java*cript

*Java*cript adalah bahasa script yang dikembangkan oleh Netscape untuk membuat dokumen yang dinamis. *Javascript* adalah bahasa script sederhana yang mempunyai kemiripan dengan bahasa pemrograman C. *Javascript* juga dikenal sebagai kode pemrograman berorientasi objek (*Object Oriented Progamming*) disingkat OOP. *Javascript* memiliki keistimewaan untuk ditambahkan pada kode HTML dan membuat dokumen menjadi lebih interaktif (Andi:2001).

2.15.2 Perbedaan *Java* dengan *Javascript*

Javascript bukan *Java*, meskipun antara keduanya ada persamaan. Bahasa *Java*cript menyerupai *Java* tetapi tidak memiliki penulisan yang statis dan kontrol yang kuat. *Javascript* mendukung banyak syntax ekspresi dan konstruksi alur kendali dasar *Java*. Perbedaannya, pada *Java* sistem waktu kompilasi pada *class* yang dibuat dari deklarasi, *Javascript* mendukung sistem *runtime* pada bilangan kecil dan tipe data yang direpresentasikan oleh tipe numerik (bilangan), *boolean* dan *string*. *Javascript* adalah sederhana yang didasari model objek. *Javascript*

juga mendukung fungsi-fungsi tanpa deklarasi khusus. Sedangkan aturan pada *Java* lebih kompleks dibandingkan dengan *Java*sript.

2.15.3 Pemakaian *Javascript*

Untuk mulai menggunakan *Javascript*, ada beberapa hal yang dibutukan oleh seorang perancang web, yaitu:

- Perancang harus mengetahui bagaimana menggunakan HTML dan mengedit dokumen HTML.
- Perancang harus menggunakan browser yang mendukung pemrograman *Javascript*, misalnya Mozila Firefox, Google Chrome, atau web brower lainnya.
- Meskipun penggunaan suatu bahasa pemrograman tidak menjadi hal yang utama, tetapi dengan mengetahui dan menguasai salah satu bahasa pemrograman akan sangat membantu dalam mempelajari *Javascript*.

Pemakaian *Javascript* dalam pembuatan web adalah dengan memasukannya dalam HTML. *Javascript* sebagai sebuah bahasa pemrograman untuk *client* dan *server* mempunyai elemen-elemen sebagai berikut:

- Kata kunci (keyword), statemen, syntax, dan *grammar*
- Aturan untuk ekspresi, variabel dan literal
- Objek dan fungsi *built-in*

2.15.4 *Javascript Grammar*

Kode *Javascript*, sebagaimana bahasa pemrograman yang lain adalah membuat statemen yang dapat melayani pembuatan proses penugasan (*assigement*

process), pembandingan nilai, dan mengeksekusi kode-kode lainnya. *Programmer* akan mudah menggunakan variabel, operator dan statemen *Java*cript elemen utama dari *Javascript grammar* adalah:

Tabel 2.2 Elemen pada *Javascript Grammar*

Variabel	Label yang nilainya dapat berubah Contoh: Total mungkin suatu saat bernilai 10
Operator	Digunakan untuk menghitung atau membandingkan nilai Contoh: Operasi penjumlahan (+), total+bonus Gaji_bersih > 100000
Ekspresi	Kombinasi sembarang dari variabel operator dan statemen yang mengevaluasi beberapa hasil Contoh: Total = 10 *if* (total > 10)
Statemen	Kumpulan semua elemen-elemen *grammar* statemen *Javascript* mungkin terdiri atas *loop*, manipulasi objek dan kondisi. Contoh: *If* (total > 10) *then* {statemen} *else* {statemen}
Objek	Susunan yang mempunyai kumpulan nilai, dimana setiap nilai merefleksikan properti setiap objeknya
Fungsi dan metodenya	Fungsi pada *Javascript* menyerupai *subroutine* atau *procedure* pada bahasa pemrograman yang lain. Sebuah fungsi adalah kumpulan statemen diskrit dengan banyak pekerjaan.

2.15.5 Memanipulasi *Window*

Alasan utama menggunakan *Javascript* untuk memanipulasi *window* (jendela), adalah karena anda dapat mengatur banyak hal pada jendela yang tidak pernah dapat dilakukan dengan HTML. *Javascript* memungkinkan anda menggunakan perintah-perintah untuk memutuskan bagian mana dari jendela *browser* yang akan tampil. Ini dilakukan menggunakan bagian ketiga dari perintah *window*.open. Bentuk penulisannya adalah

window.open ('link.html' , windowku', 'fitur-fitur window'):

Terdapat beberapa properti yang dapat diset. Misalnya, anda ingin jendela yang hanya mempunyai *location* bar dan status bar maka anda menulis kode *Javascript* berikut:

window.open ('link.html' , windowku', 'location, status'):

Berikut ini adalah beberapa fitur atau properti yang dimiliki oleh jendela web browser yang dapat diset oleh *Javascript*:

Tabel 2.3 Properti pada Jendela Web *Browser* Oleh *Javascript*

Properti	**Keterangan**
Menubar	Menampilkan menu *File*, *Edit* dan lain-lain terdapat pada bagian atas *browser*
Scroolbar	Menampilkan scrollbar pada bagian samping jendela jika diperlukan
Width	Anda dapat mengatur lebar jendela dalam ukuran *pixed*
Height	Anda dapat mengatur tinggi jendela dalam ukuran *pixed*
Toolbar	Ini akan menampilkan *toolbar browser* dengan tombol *Back*, *Stop*, *Refresh*, dan lain-lain.
Location	*Location* bar tempat anda menuliskan URL (alamat web site) akan ditampilkan di *browser*
Resizableq	Ini akan memungkinkan jendela diubah ukurannya oleh pengunjung
Directories	Ini akan menampilkan direktori-direktori pada web *browser*

UNNES
UNIVERSITAS NEGERI SEMARANG

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan pembahasan pencarian rute terpendek pendistribusian air minum isi ulang di *home industry* Rizqua di kabupaten Pekalongan di atas dapat ditarik kesimpulan sebagai berikut.

1. Berdasarkan penelitian penerapan dengan algoritma Greedy dan algoritma A* diperoleh 4 rute yang berbeda, yaitu
   a. Algoritma A* untuk mengetahui jarak yang dapat dilalui agar memperoleh jarak terpendek tanpa mempedulikan kondisi kepadatan lalu lintas. Hal ini memperoleh panjang rute 19,55 Km. Rute tersebut memiliki kepadatan lalu lintas yang besar sehingga meskipun dengan jarak terpendek, tetapi waktu yang ditempuh cenderung lebih lama.
   b. Algoritma A* untuk mengetahui kepadatan lalu lintas agar memperoleh efisiensi waktu tempuhnya dengan mempedulikan kondisi jaraknya. Hal ini memperoleh panjang rute 27,80 Km. Rute tersebut memiliki kepadatan lalu lintas yang kecil sehingga meskipun dengan jarak yang jauh, tetapi waktu yang ditempuh lebih efisien.
   c. Algoritma Greedy untuk mengetahui jarak yang dapat dilalui agar memperoleh jarak terpendek tanpa mempedulikan kondisi kepadatan lalu lintas. Hal ini memperoleh panjang rute 23,95 Km. Rute tersebut

memiliki kepadatan lalu lintas yang besar sehingga meskipun dengan jarak terpendek, tetapi waktu yang ditempuh cenderung lebih lama.

d. Algoritma Greedy untuk mengetahui kepadatan lalu lintas agar memperoleh efisiensi waktu tempuhnya dengan mempedulikan kondisi jaraknya. Hal ini memperoleh hasil panjang rute 28,7 Km. Rute tersebut memiliki kepadatan lalu lintas yang kecil sehingga meskipun dengan jarak yang jauh, tetapi waktu yang ditempuh lebih efisien.

Bisa dilihat dari hasil perhitungannya, bahwa rute dengan mempedulikan kondisi kepadatan lalu lintas maka akan menghasilkan rute yang ditempuh akan semakin besar. Jadi dengan menghasilkan keempat rute tersebut, maka bisa menjadi bahan pertimbangan dari *home industry* tersebut dalam pendistribusian air minum isi ulang tersebut.

2. Rute yang dihasilkan algoritma A* dapat menyelesaikan masalah pendistribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan, karena panjang rute yang dihasilkan lebih pendek daripada rute dari *home industry* tersebut. Sedangkan algoritma Greedy tidak dapat menyelesaikan masalah pendistribusian air minum isi ulang dari *home industry* Rizqua di Kabupaten Pekalongan, karena panjang rute yang dihasilkan lebih panjang daripada rute dari *home industry* tersebut.

   Yang artinya hasil penelitian menggunakan algoritma A* dapat membantu menyelesaikan masalah pendistribusian air minum isi ulang di *home industry* Rizqua ini.

UNNES
UNIVERSITAS NEGERI SEMARANG

3. Dalam penelitian ini, algoritma A* lebih baik daripada algoritma Greedy untuk menyelesaikan masalah rute dan kepadatan lalu lintas. Karena pada nilai yang dihasilkan algoritma A* cenderung memiliki rute yang terpendek daripada algoritma Greedy.

5.2 Saran

1. Penelitian ini diharapkan bisa menjadi bahan pertimbangan *home industry* Rizqua terkait pendistribusian air minum isi ulang di Kabupaten Pekalongan agar bisa menjadi lebih optimal.
2. Tidak semua permasalahan pencarian rute efisien pendistribusian dapat diselesaikan dengan algoritma Greedy dan algoritma A*, kedua algoritma ini masih mempunyai kelemahan karena tidak ada jaminan bahwa perhitungan yang sudah diperoleh dengan menggunakan algoritma Greedy dan algoritma A* sudah optimal.
3. Perhitungan rute terpendek pendistribusian air minum isi ulang dari *home industry* Rizqua dengan menggunakan algoritma Greedy tidak cenderung cocok untuk menyelesaikan masalah pendistribusian dengan simpul yang banyak, dan algoritma Greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada.
4. Rute yang dihasilkan algoritma A* lebih pendek daripada algoritma Greedy, sehingga sebaiknya *home industry* Rizqua memilih rute yang dihasilkan algoritma A*.
5. Perhitungan rute terpendek pendistribusian air minum isi ulang dari *home industry* Rizqua dengan menggunakan algoritma A* juga memiliki

kekurangan yaitu membutuhkan waktu yang semakin lama jika daerah yang ditempuh semakin banyak pula.

6. Rute yang dihasilkan adalah rute pengiriman awal air minum isi ulang *home industry* Rizqua Kabupaten Pekalongan, dengan kata lain kedelapan daerah pengiriman dilewati.

UNNES
UNIVERSITAS NEGERI SEMARANG

DAFTAR PUSTAKA

____ 1997. *Manual Kapasitas Jalan Indonesia* (MKJI). Departemen Pekerjaan Umum Direktorat Jenderal Bina Marga.

Andi & Wahana Komputer. 2001. *Panduan Praktis Pengembangan Web Berbasis JavaScript dan CGI.* Yogyakarta: Andi.

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

Cahyaningsih, I., Rochmad, & Supriyono. 2013. Penggunaan Algoritma Semut Dalam Travelling Salesman Problem (TSP) Pada PT. Yamaha Agung Motor Semarang. *UNNES Journal of Mathematics,* 2(2): 121-126.

Dimyati, T. T. & A. Dimyati. 1999. *Operations Research Model-Model Pengambilan Keputusan*. Bandung: Sinar Baru Algensindo.

Hayati, E. N. & A. Yohanes. 2014. Pencarian Rute Terpendek Menggunakan Algoritma Greedy. *Seminar Nasional IENACO*.

Hillier, S. F. & J. G. Lieberman. 1990. *Pengantar Riset Operasi*. Jakarta: Erlangga.

Kreshna, J. A. 2011. Desain Rute Terpendek untuk Distribusi Koran dengan Algoritma Ant Colony System. SNTIKI III.

Mulyono, S. 2004. *Riset Operasi*. Jakarta : Lembaga Penerbit Fakultas Ekonomi UI.

Munir, R. 2001. *Matematika Diskrit*. Bandung: Informatika Bandung.

Mutakhiroh, I., F. Saptono., N. Hasanah., & R. Wiryadinata. 2007. Pemanfaatan Metode Heuristik dalam Pencarian Jalur Terpendek dengan Algoritma Semut dan Algoritma Genetika. *Seminar Nasional Aplikasi Teknologi Informasi*.

Russel S. & P. Norvig. 1995. Articial Intelligence. A Modern Approach. Prentice Hall Series in Artificial Intelligence.

So, I. G., H. Sarjono., & R. T. Herman. 2013. Penerapan Metode Hungaria Pada Perusahaan Jasa (Kasus Minimum). *BINUS BUSINNES REVIEW*, 4(2): 812-820.

Suharto, I., B. Girisuta., & A. Miryanti. 2004. *Perekayasan Metodologi Penelitian*. Yogyakarta: Andi.

Supriyani. 2008. *Analisis Perancangan Sistem Informasi Geografi untuk Pencarian Rute Terpedek Kunjungan ke Sekolah-sekolah Menengah Umum dan Kejuruan di Jakarta Barat pada Universitas Bina Nusantara.* Undergraduate Thesis. Jakarta: BINUS.

Taha, A. Hamdy. 1997. *Riset Operasi*. Jakarta: Binarupa Aksara.

Wicaksana, D. A., Alamsyah., & Z. Abidin. 2014. Solusi Travelling Salesman Problem Menggunakan Algoritma Fuzzy Evolusi. *UNNES Journal of Mathematics*, 3(1): 39-43.

Wiyanti, D. T. 2013. Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem. *Jurnal Transformatika*, 1(11): 1-6.

Zulfikar, N. 2008. *Aplikasi Algoritma Genetika untuk Mencari Rute Terpendek N-Buah Node*. Skripsi. FTIK Unikom.

UNNES
UNIVERSITAS NEGERI SEMARANG

Lampiran 1