

Penerapan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek pada Pemodelan Jaringan Pariwisata di Kota Semarang

Friska Widya Ningrum¹ dan Tatyantoro Andrasto²

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Indonesia
friskawidyaningrum2@gmail.com¹, tatyantoro@mail.unnes.ac.id²

Abstrak— Kota Semarang merupakan kota yang berpotensi untuk dikembangkan menjadi daerah tujuan wisata. Informasi tentang obyek wisata sangat dibutuhkan oleh para wisatawan salah satunya adalah informasi rute wisata terpendek menuju beberapa obyek wisata populer di Kota Semarang, sehingga dapat mengefisiensi waktu, jarak, dan biaya. Tujuan dari penelitian ini adalah menerapkan algoritma Floyd-Warshall dalam mencari jalur terpendek jaringan pariwisata Kota Semarang. Peta Kota Semarang ditransformasikan kedalam bentuk diagram grafik. Algoritma Floyd-Warshall diterapkan dalam perhitungan bobot *path* dari diagram grafik untuk mencari rute terpendek ke obyek wisata populer di Kota Semarang. Sistem pencarian rute terpendek pariwisata Kota Semarang ini dapat menjadi media promosi pariwisata Kota Semarang dan dapat dimanfaatkan sebagai alternatif rute perjalanan oleh wisatawan.

Kata kunci : Pariwisata Kota Semarang, Algoritma Floyd-Warshall, Rute Terpendek

I. PENDAHULUAN

Kota Semarang merupakan kota yang berpotensi untuk dikembangkan menjadi daerah tujuan wisata. Kota Semarang juga salah satu daerah tujuan wisata, baik wisatawan domestik maupun mancanegara, sehingga industri pariwisata Kota Semarang perlu adanya pengembangan, salah satunya dengan dukungan teknologi informasi. Penerapan teknologi informasi dalam dunia pariwisata dapat diterapkan baik untuk meningkatkan promosi dan penyampaian informasi maupun meningkatkan kualitas pelayanan obyek wisata. Salah satu contoh penerapan teknologi informasi dalam meningkatkan informasi pariwisata adalah adanya aplikasi yang memberikan informasi mengenai rute berbagai obyek wisata.

Wisatawan yang datang berkunjung membutuhkan informasi rute wisata untuk membantu merencanakan perjalanan selama berwisata hingga ke tempat tujuan yaitu obyek wisata yang dituju dan dari obyek wisata hingga kembali ke tempat tinggal asal maupun tempat sementara [1]. Selain itu wisatawan juga mencari rute terpendek menuju tempat-tempat wisata yang akan dikunjungi agar dapat mengefisiensi waktu, jarak, dan biaya [2].

Pencarian rute terpendek dapat dicari dengan menggunakan algoritma grafik, salah satunya menggunakan algoritma *Floyd-Warshall*. Algoritma *Floyd-Warshall* adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu.

Algoritma yang ditemukan oleh Warshall untuk mencari rute terpendek merupakan algoritma yang sederhana dan mudah implementasinya [3]. Algoritma *Floyd-Warshall*

memiliki input graf berarah dan berbobot (V,E), yang berupa daftar titik (*node*/titik V) dan daftar sisi (sisi E). Bobot garis e dapat diberi simbol $w(e)$. Jumlah bobot sisi-sisi pada sebuah jalur adalah total bobot jalur tersebut. Sisi pada E diperbolehkan memiliki bobot negatif, akan tetapi tidak diperbolehkan bagi graf W_{ij} untuk memiliki siklus dengan bobot negatif. Algoritma ini menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang paling minimum.

Algoritma *Floyd-Warshall* untuk mencari lintasan 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) $W^* = W$.

Keterangan :

W_0 = matriks keterhubungan graf berarah berbobot awal

W^* = matriks keterhubungan minimal

$W_{i,j}$ = lintasan terpendek dari titik v_i ke v_j

Dalam iterasinya untuk mencari lintasan terpendek, algoritma *Floyd-Warshall* membentuk n matriks, sesuai dengan iterasi- k . Ini akan menyebabkan prosesnya lambat, terutama untuk nilai n yang besar. Meskipun waktu prosesnya bukanlah yang tercepat, algoritma *Floyd-Warshall* sering dipergunakan untuk menghitung lintasan terpendek karena kesederhanaannya. Di samping itu, implementasi algoritma *Floyd-Warshall* sangat mudah dibuat.

Matriks keterhubungan W yang digunakan untuk menyatakan graf berarah berbobot sama dengan matriks yang digunakan untuk menyatakan graf berbobot, yaitu elemennya menyatakan bobot garis. Secara umum matriks keterhubungan untuk menyatakan graf berarah berbobot tidaklah simetris karena bobot garis dari titik v_i ke v_j ($W_{i,j}$) tidak sama dengan bobot garis dari titik v_j ke v_i ($= W_{j,i}$) dan $W_{i,i} = \infty$ untuk semua i .

Algoritma *Floyd-Warshall* di atas hanya menghitung jarak terpendek dari semua titik ke semua titik, tetapi tidak menjelaskan bagaimana *path* terpendeknya. Untuk menentukan *path* yang menghasilkan jarak terpendek, maka harus ditambahkan matriks bujur sangkar Z (ukuran $n \times n$) yang disusun sebagai berikut:

$$\text{Inisialisasi } Z^{(0)}_{i,j} = \begin{cases} j & \text{jika } W_{i,j}^{(0)} \neq \infty \\ 0 & \text{jika } W_{i,j}^{(0)} = \infty \end{cases}$$

Dalam iterasi ke- k , apabila titik v_k disisipkan antara titik- i dan titik- j (berarti menukar $W_{i,j}$ dengan $W_{i,k} + W_{k,j}$), maka ganti $Z_{i,j}$ dengan $Z_{i,k}$. Agar lebih efisien, penggantian matriks Z dilakukan bersama-sama dengan iterasi pencarian jarak terpendeknya.

Revisi algoritma *Floyd-Warshall* dengan melibatkan *path* terpendeknya adalah sebagai berikut:

- 1) $W = W_0$; $Z = Z_0$
- 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
 a. Tukar $W_{[i,j]}$ dengan $W_{[i,k]} + W_{[k,j]}$.
 b. Ganti $Z_{i,j}$ dengan $Z_{i,k}$
- 3) $W^* = W$.

Keterangan :

W_0 = matriks keterhubungan graf berarah berbobot awal

W^* = matriks keterhubungan minimal

$W_{i,j}$ = lintasan terpendek dari titik v_i ke v_j

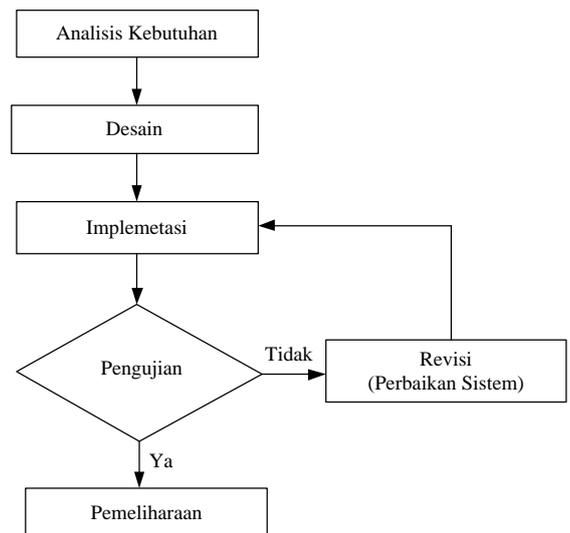
Tujuan penelitian ini, menerapkan algoritma *Floyd-Warshall* dalam menentukan rute terpendek pada pemodelan jaringan di Kota Semarang.

II. METODE

Metode pengumpulan data yang digunakan dalam penelitian ini adalah observasi. Observasi dilakukan untuk mengumpulkan data sekunder berupa obyek-obyek wisata populer di Kota Semarang dan mengamati jalur-jalur serta menentukan jarak setiap titik jaringan melalui *Google Maps*.

Pada tahap pengembangan sistem rute terpendek, penelitian ini menggunakan metode *waterfall*. *Waterfall model* adalah suatu proses pengembangan perangkat lunak berurutan, dimana kemajuan suatu sistem dipandang sebagai suatu hal yang harus mengalir kebawah (seperti air terjun) melewati fase perencanaan, pemodelan, implementasi dan pengujian [4]. Tahapan model *waterfall* ditunjukkan pada Gambar 1.

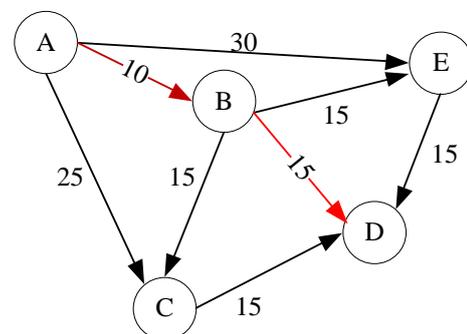
Pada Gambar 1 tahap-tahap pengembangan sistem dari menganalisis kebutuhan yang meliputi, *hardware* dan *software*. Yang kedua tahap desain, tahap desain dilakukan perancangan *Data Flow Diagram* yang bertujuan untuk memudahkan pemahaman terhadap sistem yang dibangun. Tahap Implementasi merupakan kegiatan untuk mengimplementasi semua rancangan yang telah dibuat. Pada tahap ini dilakukan proses pemrograman dari algoritma *Floyd-Warshall* dengan menggunakan bahasa PHP.



Gambar 1. Tahapan metode *Waterfall* [4]

Algoritma *Floyd-Warshall* disini membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal.

Salah satu contoh kita berada dari suatu tempat yang berada di titik D dimana kita harus melewati minimal satu titik, titik antara B, C, dan E, model jaringan ditunjukkan pada Gambar 2.



Gambar 2. Contoh implementasi jalur pada *node*

Apabila kita memakai algoritma *Floyd-Warshall* maka ada beberapa tahapan kerja dari algoritmanya yaitu :

1. Mencari *node* mana saja yang bisa dilalui untuk menuju ke *node* tujuan atau D

- Menjumlahkan nilai *edge* pada *node* dengan *edge* pada *node* yang akan dilalui mulai dari *node* awal menuju *node* tujuan.

$$A-E-D = 30+15 \text{ km}$$

$$A-B-E-D = 10+15+15 \text{ km}$$

$$\mathbf{A-B-D = 10+15 \text{ km}}$$

$$A-B-C-D = 10+15+15 \text{ km}$$

$$A-C-D = 25+15 \text{ km}$$

- Mencari nilai terkecil dari hasil penjumlahan *edge* pada *node-node* yang bisa dilalui. Dari hasil penjumlahan di atas didapat nilai terkecilnya yaitu pada jalur A-C-E dengan jumlah total *edge* 25 km.

Tahap pengembangan sistem pencarian rute terpendek selanjutnya adalah pengujian. Pengujian dilakukan untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan.

Pengujian yang dilaksanakan pada penelitian ini dibagi menjadi 3, yaitu :

- Pengujian *Black Box*, pengujian *Black Box* pengujian dalam aspek fundamental sistem tanpa memerhatikan struktur logika internal perangkat lunak. Pengujian *Black Box* cenderung mementingkan apakah sistem memberikan keluaran (*output*) seperti yang diharapkan sistem atau tidak, tanpa mengetahui sistem kerja dari perangkat lunak yang diuji.
- Pengujian hasil perhitungan, pengujian dilakukan dengan membandingkan perhitungan matriks W_{ij} Awal yang dihasilkan oleh sistem dengan matriks W_{ij} Awal yang dihitung secara manual, dalam hal ini hanya diambil potongan matriks sebesar 5x5. Jika hasil matriks W^* Akhir yang diperoleh dari sistem dengan matriks W^* Akhir yang dihitung secara manual, maka sistem sudah berjalan sesuai dengan yang diharapkan.

Kebenaran hasil jalur dan jarak tempuh yang dilewati, pengujian bagian ketiga ini berfungsi untuk mengetahui kebenaran hasil jalur dan jarak tempuh yang dihasilkan sistem. Pengujian dilakukan dengan cara melewati dan mengamati jaringan pariwisata Kota Semarang serta dibandingkan jalur hasil perhitungan sistem.

III. HASIL DAN PEMBAHASAN

Tahap awal dari pencarian rute terpendek yaitu menstranformasikan peta Kota Semarang ke dalam bentuk grafik yang hasilnya dapat dilihat pada gambar 3. Titik berwarna merah merupakan titik awal yang berupa Terminal, Stasiun dan Bandara Penggaron, titik berwarna hijau

merupakan obyek wisata serta titik berwarna biru persimpangan jalan. Pada Gambar 3 membentuk jaringan pariwisata Kota Semarang, yang terdiri dari 116 titik/*node*, yang terdiri 4 terminal, 2 stasiun, 1 bandara, 87 persimpangan jalan, 24 obyek wisata yang diwakili 22 *node*, dimana Tugumuda dan Lawang Sewu serta Taman Budaya Raden Saleh dan Wonderia masing-masing dijadikan satu titik *node* dikarenakan letak obyek wisata yang berdekatan.

Titik-titik yang ada kemudian saling dihubungkan sesuai dengan jalan yang dilewati oleh kendaraan pribadi terutama yang dilewati mobil. Ada 178 sisi yang terbentuk yang menajdi suatu jaringan pariwisata Kota Semarang.

Setelah mendapatkan *node* dan *path*, maka didapat matriks W_{ij} Awal dan matriks Z_{ij} Awal. Matriks W_{ij} Awal merupakan matriks yang berguna untuk mencari jarak terpendek setiap pasangan titik, sedangkan matriks Z_{ij} Awal berguna untuk mencari *path* terpendek antar titik.

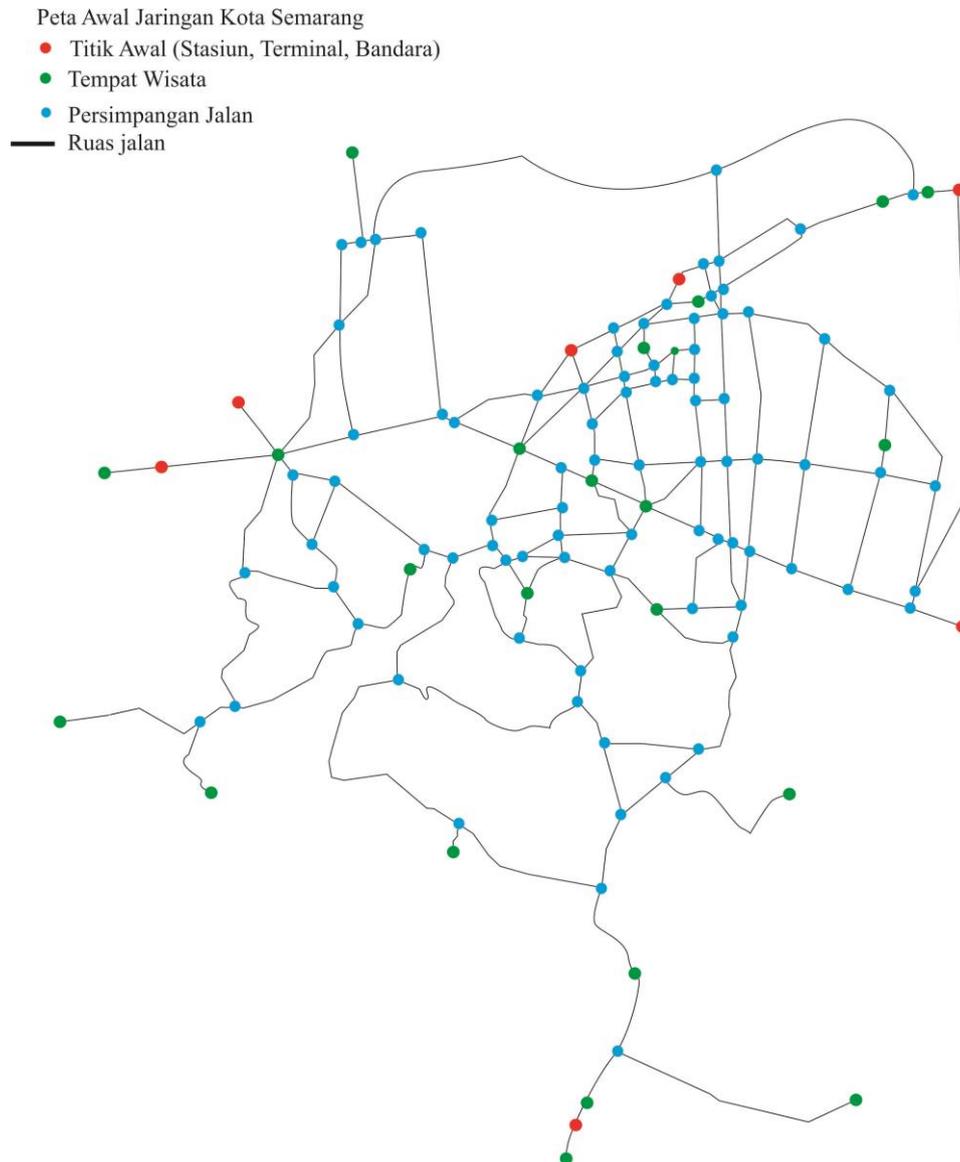
Matriks W_{ij} Awal dan Z_{ij} Awal kemudian dilakukan proses perhitungan dengan algoritma *Floyd-Warshall* untuk mencari jalur terpendek antar semua titik dalam jaringan tersebut. Setelah perhitungan dilakukan hingga iterasi terakhir, yaitu saat $k = 116$, $i = 116$, dan $j = 116$, maka didapatkan matriks W^* Akhir atau matriks akhir hasil dari proses perhitungan algoritma *Floyd-Warshall* yang menunjukkan jalur terpendek antar setiap titik dan matriks Z^* Akhir.

Berdasarkan hasil perhitungan matriks, maka dapat mengetahui jalur terpendek yang ditempuh untuk obyek wisata yang ada di Kota Semarang. Hasil yang didapat dari hasil perhitungan menggunakan algoritma *Floyd-Warshall* memberikan pilihan rute perjalanan yang lebih optimum dari segi jarak.

Hasil pengujian dengan metode *Black Box* menunjukkan fungsionalitas sistem 100% berjalan dengan baik. Untuk pengujian kedua berfungsi memeriksa hasil perhitungan matriks W_{ij} Akhir dan Z^* Akhir yang dihasilkan oleh sistem.

Hasil perbandingan matriks W_{ij} Akhir dan Z^* Akhir yang dihasilkan oleh sistem dengan perhitungan secara manual, adalah sama. Dan tahap terakhir dilakukan pengujian untuk menguji kehandalan sistem yang telah dibuat, hasil uji kehandalan sistem menunjukkan kebenaran bekerja dengan baik dan dapat diterima serta tidak terjadi penyimpangan jalur, sehingga hasil yang diharapkan sudah sesuai dengan yang diharapkan.

Dari segi pemanfaatannya sistem pariwisata Kota Semarang dimaksudkan dapat menjadi media promosi oleh Dinas Kebudayaan dan Pariwisata Kota Semarang dan dapat dimanfaatkan sebagai jalur alternatif jalur terpendek untuk rute perjalanan pariwisata oleh wisatawan.



Gambar 3. Representasi peta Kota Semarang ke dalam bentuk graf jaringan

IV. SIMPULAN DAN SARAN

Kesimpulan dari penelitian ini adalah dihasilkan aplikasi pencarian jalur terpendek menggunakan algoritma Floyd-Warshall dapat digunakan di Kota Semarang. Pencarian rute terpendek pariwisata Kota Semarang dimulai dari menstransformasikan peta Kota Semarang ke dalam bentuk jaringan pariwisata Kota Semarang yang memuat terminal, stasiun dan bandara sebagai titik atau *node* awal dan tempat wisata sebagai titik tujuan, sedangkan persimpangan jalan menghubungkan antar titik atau *node*. Kemudian jaringan yang telah terbentuk dibuat matriks ketetanggaan dan dihitung menggunakan algoritma Floyd-Warshall hingga didapatkan pilihan jalur yang terpendek.

REFERENSI

- [1] Manongga, D., Papilaya, S., & Pandie, S. 2009. Sistem Informasi Geografis untuk Perjalanan Wisata di Kota Semarang. Jurnal Informatika Vol.10, No.1, 1-9.
- [2] Liwang, R., Santoso, A. J., & Rahayu, F. S.. 2013. Rancang Bangun Aplikasi Menggunakan Floyd Warshaall. Seminar Nasional Teknologi Informasi dan Multimedia (pp. 18-21). Yogyakarta : STMIK AMIKOM Yogyakarta.
- [3] Siang, Jong Jek. 2002. Matematika Diskrit dan Aplikasinya pada Ilmu Komputer. Yogyakarta : ANDI
- [4] Pressman, R. S. 2002. Rekayasa Perangkat Lunak Pendekatan Praktis Buku I. Yogyakarta : ANDI