

PEMANFAATAN METODE *ADJACENCY MATRIX* UNTUK OPTIMASI RUTE JALAN BERBASIS WEB

Deni Ramdan¹, Galih Hermawan²

^{1,2}Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112-116 Bandung

ABSTRAK

Seiring dengan perkembangan jaman infrastruktur jalan banyak dibangun, banyak rute-rute baru yang dapat dilalui oleh alat transportasi umum saat ini, lahirlah masalah-masalah baru seperti rute mana yang merupakan rute terpendek dan berapa banyak kemungkinan rute yang bisa dilalui, hal ini mengharuskan pelaku perjalanan transportasi umum berfikir pintar. Dalam makalah penelitian ini disajikan hasil penelitian kami dalam mengembangkan sebuah aplikasi berbasis web untuk mensimulasikan optimasi rute jalan menggunakan pendekatan *adjacency matrix*.

Hasil penelitian menunjukkan bahwa penerapan metode *adjacency matrix* dalam optimasi rute jalan yang diterapkan dalam aplikasi dapat membantu pengguna untuk mengelola data rute jalan, yang selanjutnya dapat digunakan untuk menemukan rute terpendek dan rute alternatif di antara dua *node*, disertai dengan informasi rute jalan berdasarkan biaya transportasi yang dimasukkan sebagai contoh data simulasi.

Kata Kunci : Rute terpendek, *adjacency matrix*, optimasi rute jalan, *graph matrix*.

1. PENDAHULUAN

Seiring dengan itu infrastruktur jalan pun banyak dibangun, hasilnya banyak rute-rute baru yang dapat dilalui oleh alat transportasi yang ada pada saat ini, seiring dengan itu pula lahirlah masalah-masalah baru seperti rute mana yang merupakan rute terpendek dan berapa banyak kemungkinan rute yang bisa dilalui, hal ini mengharuskan pelaku perjalanan berkendara umum untuk berfikir pintar. Untuk pelaku perjalanan yang belum mengetahui rute tujuan, hal tersebut bisa menjadi masalah karena mengambil rute yang salah, seperti contoh untuk menuju ke tujuan B dari tujuan A harusnya jaraknya 50 Km dengan tidak mengetahui rute bisa menjadi 100 Km jarak perjalanan.

Oleh karena itu dibutuhkan analisis bagaimana menentukan rute terpendek dari alternatif rute yang ada dan kemungkinan alternatif rute yang bisa

dilalui untuk menuju ke tujuan, serta semua alternatif yang bisa dilalui ke semua tujuan yang masuk dalam jaringan rute.

Salah satu metode yang digunakan untuk optimasi rute jalan adalah *adjacency matrix*, berikut dikembangkan sebuah aplikasi berbasis web untuk mensimulasikan optimasi rute jalan yang telah dibuat.

2. TINJAUAN PUSTAKA

Matrix

Matrix adalah suatu kumpulan besaran (variabel dan konstanta) yang dapat dirujuk melalui indeksnya, menyatakan posisinya dalam representasi umum yang digunakan, yaitu sebuah tabel persegi panjang. *Matrix* merupakan suatu cara visualisasi variabel yang merupakan kumpulan dari angka-angka atau variabel lain, misalnya vektor. Dengan representasi *matrix*, perhitungan dapat dilakukan dengan lebih terstruktur. Pemanfaatannya misalnya dalam menjelaskan persamaan linier, transformasi koordinat, dan lainnya. *Matrix* seperti halnya variabel biasa dapat dimanipulasi, seperti dikalikan, dijumlah, dikurangkan dan didekomposisikan. Contoh persamaan *matrix* dapat dilihat pada persamaan (P.1).

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \dots (P.1)$$

A. Teori Matrix Grafik

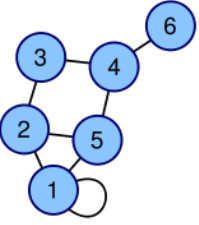
Berikut ini akan dibahas mengenai teori *matrix* grafik.

1. *Adjacency Matrix* adalah sebuah *matrix* bujur sangkar yang mewakili suatu grafik, dengan aij bukan nol jika vertex i dan vertex j adalah bersebelahan.
2. *Biadjacency Matrix* adalah sebuah kelas khusus mengenai *matrix adjacency* yang menguraikan grafik dua partai.

3. *Degree Matrix* adalah sebuah diagonal *matrix* yang menggambarkan tingkat derajat vertex dalam suatu grafik.
4. *Incidence Matrix* adalah sebuah *matrix* menggambarkan sebuah hubungan antara 2 kelas objek (biasanya puncak dan ujung dari konteks grafik).
5. *Laplacian Matrix* adalah sebuah *matrix* yang mirip dengan *degree matrix* kurang *adjacency matrix* untuk sebuah grafik yang digunakan untuk menemukan banyaknya pohon dalam grafik.
6. *Seidel Adjacency Matrix* adalah sebuah *matrix* serupa dengan *adjacency matrix* tetapi dengan -1 adjacency, +1 untuk bukan adjacency, 0 dalam diagonal.

B. Adjacency Matrix

Pada ilmu matematika dan ilmu komputer, *adjacency matrix* pada grafik G terarah atau tidak terarah, dalam n dimana n x n matrix non diagonal a ij adalah nomor dari vertex i ke j, dan diagonal a ij adalah dua kali perulangan dalam vertex i atau hanya sekali perulangan (pemakaian berbeda, tergantung dalam keperluan matematikal, artikel ini mengarahkan konvensi terdahulu untuk grafik tidak terbatas. Meskipun grafik terarah selalu mengikuti yang terakhir). Terdapat sebuah *adjacency matrix* unik untuk grafik dan matrix ini bukan *adjacency matrix* dalam beberapa grafik. Dalam kasus tertentu mengenai grafik sederhana terbatas, *adjacency matrix* adalah sebuah (0,1) - *matrix* dengan nol dalam diagonal. Pada Gambar. 1 berikut adalah contoh ilustrasi *adjacency matrix*.

Grafik	Adjacency Matrix
	1 1 0 0 1 0
	1 0 1 0 1 0
	0 1 0 1 0 0
	0 0 1 0 1 1
	1 1 0 1 0 0
	0 0 0 1 0 0

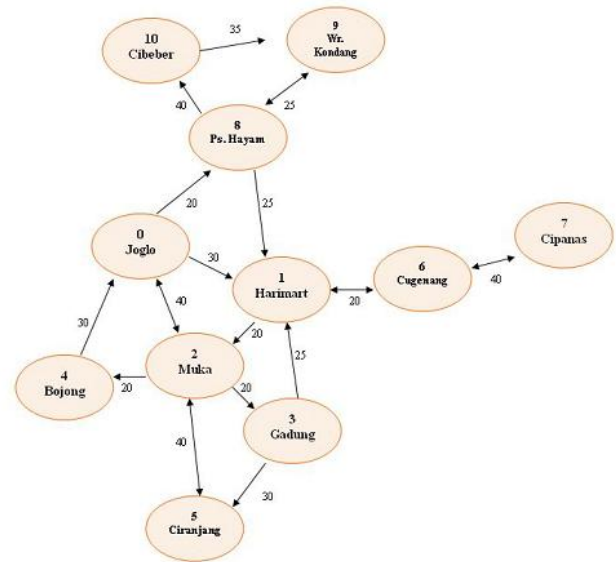
Gambar. 1 Adjacency matrix

3. ANALISIS DAN PERANCANGAN

A. Mekanisme Alokasi Rute

Sebuah rute tidak lepas dari faktor-faktor seperti tempat, alat transportasi dan jarak, ketiga faktor tersebut harus dapat diidentifikasi agar tercipta sebuah informasi rute yang diharapkan. Pada kasus ini penulis mencoba menerapkan kasus untuk rute dengan menggunakan kendaraan umum, sehingga untuk menuju ke suatu tujuan memungkinkan menggunakan beberapa kendaraan umum.

Dari ketiga faktor yang telah disebutkan penulis menggunakan data tempat sebagai data utama dan faktor alat transportasi dan jarak dijadikan data pendukungnya. Untuk memudahkan dalam menghubungkan data dengan metode matrix, data tempat harus diidentifikasi lagi menjadi kode-kode berupa angka.



Gambar. 2 Ilustrasi rute jalan

Pada Gambar. 2 tersebut menunjukkan contoh ilustrasi rute jalan dalam sebuah daerah yang di dalamnya terdapat *node-node* sub daerah yang saling terhubung disertai angka yang menunjukkan jarak antara *node-node* yang berhubungan tersebut.

C. Mekanisme Konfigurasi Rute

Aplikasi Rute untuk dapat menyajikan informasi rute dan informasi layanan – layanan jaringan lainnya harus melalui proses konfigurasi terlebih dahulu. Konfigurasi dapat dibuat sesuai kebutuhan di dalam suatu jaringan rute. Pertimbangan dalam analisis rute sebaiknya berdasarkan faktor - faktor yang tersebut.

Konfigurasi analisis rute berisikan pendefinisian parameter-parameter informasi rute seperti berikut ini.

1. Pendefinisian tempat atau lokasi.
2. Pendefinisian jarak dari satu titik ke titik lainnya.
3. Pendefinisian alat transportasi yang digunakan.
4. Pendefinisian jalur atau trayek yang dapat melewati sebuah jalur.

Pendefinisian parameter-parameter informasi yang dideklarasikan dalam konfigurasi rute tersebut, selanjutnya akan dimasukkan ke dalam teori *matrix* dan menghasilkan pendekatan informasi-informasi rute seperti berikut ini.

1. Menentukan rute terpendek.
2. Kemungkinan rute alternatif yang bisa dilalui untuk menuju suatu tujuan.

3. Kemungkinan kendaraan umum yang dinaiki untuk menuju suatu tujuan.

4. IMPLEMENTASI DAN PENGUJIAN

A. Penetapan Rute Jalan Dalam Sebuah Matrix

Node-node yang merepresentasikan daerah atau tempat disimbolkan sebagai indeks baris dan kolom dalam sebuah *matrix*. Dimana jika asumsi terdapat 6 buah tempat, maka ilustrasi *matrix* adalah sebagai berikut.

```

10 $m = array( // 0 1 2 3 4 5
11     array( 0, 0, 0, 0, 0, 0), // 0
12     array( 0, 0, 0, 0, 0, 0), // 1
13     array( 0, 0, 0, 0, 0, 0), // 2
14     array( 0, 0, 0, 0, 0, 0), // 3
15     array( 0, 0, 0, 0, 0, 0), // 4
16     array( 0, 0, 0, 0, 0, 0), // 5
17 );
    
```

Gambar. 3 *Matrix* untuk menyimpan data rute, dimana masing-masing tempat masih belum ada hubungan

Dalam aplikasi yang dibuat telah disediakan sebuah mekanisme untuk pengelolaan data tempat berikut data rute yang menghubungkan tempat-tempat yang telah dibuat dan tersimpan dalam database.

Dalam Gambar. 4 telah tersedia contoh tampilan untuk mengelola data rute berdasarkan tempat-tempat yang ada.

Gambar. 4 Antarmuka pengelolaan data rute

Apabila kita melihat desain awal penyesuaian data-data rute yang ada dalam database ke sebuah *matrix* sebagaimana terlihat pada Gambar. 3 sebelumnya, ilustrasi *matrix* data rute lengkapnya adalah sebagai berikut.

```

10 $m = array( // 0 1 2 3 4 5
11     array( 0,30, 40, 0, 0, 0), // 0
12     array( 0, 0, 20, 0, 0, 0), // 1
13     array( 40, 0, 0, 20,20,40), // 2
14     array( 0, 0, 0, 0, 0,30), // 3
15     array( 30, 0, 0, 0, 0, 0), // 4
16     array( 0, 0,40, 0, 0, 0), // 5
17 );
18
    
```

Gambar. 5 Ilustrasi *matrix* dengan data rute lengkap

B. Pengelolaan Transportasi

Data transportasi dalam hal ini digunakan untuk mewakili alat-transportasi apa saja yang dapat digunakan nantinya dalam rute-rute yang sudah dibuat. Antarmukanya dapat dilihat pada Gambar. 6 berikut ini.

Gambar. 6 Antarmuka pengelolaan data transportasi

C. Pengelolaan Ongkos Transportasi Dalam Rute

Sebagai penunjang kelengkapan informasi transportasi dalam aplikasi ini, tersedia juga antarmuka untuk penetapan alat transportasi apa yang dapat digunakan dalam sebuah rute yang pernah dibuat, berikut dengan ongkos yang diperlukan.

Gambar. 7 Antarmuka pengelolaan data alokasi alat transportasi berdasarkan rute

D. Simulasi Rute Alternatif

Pada simulasi ini akan dicoba memilih rute dari Joglo menuju Muka, sebagaimana dapat dilihat pada gambar 7.



Gambar 7. Antarmuka simulasi rute alternatif

Dalam simulasi rute alternatif tersebut, dari hasil pencarian telah ditemukan 2 buah rute yang dapat dilalui, yaitu :

1. Joglo menuju Harimart dan sampai di muka, dengan jarak 50 KM.
2. Joglo menuju Muka langsung dengan jarak 40 KM.

Hasil pencarian rute di atas sama dengan yang tertera pada Gambar. 2 mengenai kasus rute. Selain nama tempat dan jarak, terdapat pula data transportasi yang dapat digunakan berikut ongkos yang harus dikeluarkan.

Rute alternatif 1 menjelaskan untuk menuju muka dari Joglo bisa dilalui dengan 05A ataupun naik 05A sampai Harimart lalu naik BUS untuk menuju ke Muka dengan ongkos dan kecepatan yang berbeda. Kesimpulan dari hasil pencarian di atas adalah jalur tercepat dan termurah dapat menggunakan rute alternatif 2.

E. Simulasi Rute Terpendek

Simulasi selanjutnya adalah untuk mencari rute terpendek. Dalam hal ini, rute yang dipilih adalah sama dengan yang digunakan dalam simulasi rute alternatif agar penelitian terhadap kasus tidak berbeda-beda.

Hasil simulasi dapat dilihat pada gambar 8



Gambar. 8 Antarmuka simulasi rute terpendek

Dalam Gambar.8 di atas dapat diperoleh informasi bahwa rute menuju Muka dari Joglo mempunyai rute terpendek 40 KM dengan menggunakan transportasi 05B dan memerlukan ongkos sebesar Rp.1.000,-

Hasil tersebut sama persis nilainya ketika melakukan pencarian rute alternatif.

F. Simulasi Rute Alternatif dengan OngkosTermurah

Pemanfaatan data rute, data transportasi, dan data ongkos di setiap transportasi pada rute yang bersangkutan selain untuk memperoleh informasi rute terpendek, juga dapat digunakan untuk memperoleh informasi rute dengan ongkos termurah.

Contoh simulasi dapat dilihat pada Gambar. 9, dimana dalam hal ini hendak dicari rute yang menghubungkan antara Joglo dengan Harimart.



Gambar. 9 Hasil simulasi pencarian rute dari Joglo ke Harimart

Secara umum, dalam aplikasi simulasi ini telah berhasil menunjukkan bagaimana memperoleh informasi rute terpendek, rute alternatif, dan rute berdasarkan ongkos termurah, disertai dengan jenis transportasi yang telah ditentukan.

5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian adalah :

1. Dalam mengelola rute jalan yang banyak dan kompleks, konfigurasi rute jalan pada aplikasi

komputer dapat membantu seorang menemukan rute jalan secara cepat. Metode *Adjacency Matrix* merupakan solusi yang tepat untuk menangani konfigurasi rute jalan tersebut. Data utama dan data pendukung rute merupakan faktor olahan yang harus dimasukkan seperti data tempat, data rute, data transportasi dan data jalur atau trayek, sehingga dapat ditemukan sebuah informasi seperti rute jalan alternatif dan rute jalan terpendek.

2. Kemudahan dalam mengatur konfigurasi rute jalan perlu diperhatikan . Fungsi dari aplikasi web yang dirancang ini dapat digunakan untuk membantu melakukan konfigurasi rute jalan. Aplikasi ini juga dapat membantu administrasi rute dalam *me-monitoring* dan menciptakan informasi rute jalan yang tepat sesuai dengan kenyataannya.

Adapun saran yang kami ajukan adalah sebagai berikut.

1. Untuk lebih meningkatkan aspek validitas dalam konfigurasi rute jalan, perlu adanya dukungan dari badan atau instansi yang berwenang dalam rute jalan sehingga dapat meminimalkan faktor kesalahan data rute.
2. Aspek sumberdaya manusia untuk administrasi rute jalan perlu diselaraskan dengan kemajuan teknologi sekarang ini, terlebih tentang aspek banyaknya rute dan kompleks.
3. Pada aplikasi yang dirancang perlu ditambahkan fungsi-fungsi yang dapat membantu administrasi dan *monitoring* rute lalu lintas jalan

DAFTAR PUSTAKA

1. Bunafid Nugroho (2007), "PHP Professional pengembangan data array dalam aplikasi Web ", Andi, Yogyakarta
2. Didik Dwi Prasetyo (2003), "Administrasi Database Server MySQL", Elex Media Komputindo, Jakarta
3. Didik Dwi Prasetyo (2004), "Solusi Pemrograman berbasis Web menggunakan PHP 5", Elex Media Komputindo, Jakarta
4. http://en.wikipedia.org/wiki/Graph_theory
5. http://en.wikipedia.org/wiki/Matrix_%28mathe%20matis%29
6. <http://www.brpreiss.com/books/opus4/html/pag e533.html>
7. http://www.cs.usask.ca/resources/tutorials/cscon cepts/1999_8/tutorial/beginner/matrices/matrix. html
8. Roger S Pressman (2002), "Rekayasa Perangkat Lunak Pendekatan Praktisi ", Andi, Yogyakarta

9. Yahya Kurniawan (2002), "Aplikasi Web Database dengan PHP dan MySQL", Elex Media Komputindo, Jakarta