PERANCANGAN APLIKASI ANALISIS RUTE JALAN DENGAN MENGGUNAKAN PENDEKATAN *ADJACENCY MATRIX* BERBASIS WEB

Deni Ramdan

Pembimbing 1 : Manahan P Siallagaan, S.Si, M.T

Pembimbing 2 : Galih Hermawan, S.Kom

Fakultas Teknik dan Ilmu Komputer, Jurusan Teknik Informatika

Universitas Komputer Indonesia

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.

Adjacency Matrix merupakan standar *graph matrix* dengan menggunakan pendekatan *array* pada *php* dengan bantuan *database MySQL* mencoba memecahkan masalah jalur transportasi umum.

Dalam memecahkan informasi rute jalan diperlukannya konfigurasi analisis rute jalan melalui sebuah aplikasi *web*. Fungsi dari aplikasi *web based* yang dirancang agar dapat mempermudah pencarian dan administrasi data rute jalan.

Kata Kunci : ***Adjacency Matrix, Graph Matrix, Array, PHP, MySQL, web***

ABSTRACT

During with infrastructure era growth of road, many route able to pass by public transportastion appliance in time. New problems raise up like when route is the shorte route and how many possibility of route which one can pass by, this matter oblige the user of public transportation think more and right.

Adjacency Matrix are graph matrix standard which is use array approach to php with database MySQL try to solve common transportation route problem.

Along try to solving route information to user it need route analysis configurations trough web application. Function of web based application as design to find route data more easier

Keyword : ***Adjacency Matrix, Graph Matrix, Array, PHP, MySQL, web***

1. Pendahuluan

Seiring kemajuan jaman, lokasi dan jarak tidak dijadikan masalah besar dalam melakukan perjalanan, ditambah lagi dengan kemajuan teknologi transportasi dari berbagai produsen kendaraan bermotor yang berupaya memuaskan konsumennya dengan kenyamanan dalam melakukan perjalanan.

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 berkendaraan 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. Berdasarkan hal diatas, penulis tertarik membuat aplikasi untuk menganalisis permasalahan tersebut dengan menggunakan pendekatan *Adjacency matrix* pada *platform* web.

2. Latar Belakang

Semakin banyak jalur alternatif, harus semakin jeli juga dalam menentukan rute yang akan dilalui, kondisi tersebut memaksa pengguna jalan bagaimana menentukan rute terpendek, menentukan rute alternatif ke suatu tujuan serta alternatif rute ke semua tujuan. Pada kenyataan dilapangan sebuah rute terpendek belum tentu menjadi solusi yang terbaik mengingat kondisi dilapangan sangat sulit untuk diprediksi. Dalam penggunaan pendekatan *Adjacency Matrix*, pemilihan input data untuk rute mempunyai peranan penting agar diperoleh hasil prediksi yang baik. Selanjutnya perumusan masalah dapat dirumuskan dalam pernyataan sebagai berikut :

1. Bagaimana cara membuat perangkat lunak untuk analisis rute jalan dengan pendekatan *Adjacency Matrix*.
2. Variable input apa saja yang dimasukan ke dalam jaringan rute.

3. *Matrix*

Pembelajaran *Matrix* telah sejak lama ada. *Latin Squares* dan *Magic Squares* telah dikenal sejak jaman prasejarah. *Matrix* mempunyai sejarah panjang tentang aplikasi dalam memecahkan persamaan linear. Beberapa huruf china penting antara *300BC* sampai *AD200*, sembilan bab dalam seni matetamika (*Chiu Chang Suan Shu*) adalah contoh pertama penggunaan metode *matrix* dalam memecahkan persamaan simultan. Pada bab ketujuh "*Too Much and Not Enough*", konsep penentu pertama terlihat hampir 2000 tahun sebelum penemuan oleh Ahli matematik Jepang *Seki Kowa* pada tahun 1683 atau *Leibniz Gottfried* Jerman (seseorang yang telah dihargai dengan penemuan tentang kalkulus/diferensial, yang terpisah namun serempak dengan *Isaac Newton*). *Leibniz* mengembangkan teori determinan pada tahun 1693. *Cramer* mengembang;kan teori tersebut lebih lanjut, mempresentasikan aturan *Cramer's* pada tahun 1750. *Carl Friedrich Gauss* dan *Wilhelm Jordan* mengembangkan *Gauss-Jordan* penghapusan pada tahun 1800. Istilah *matrix* yang pertama tahun 1848 dengan *J. J. Sylvester. Cayley, Hamilton, Grassmann, Frobenius* dan *Von Neumann* adalah antara para ahli matematik yang sudah terkenal.

3.1 Pengenalan *Matrix*

Matrix adalah suatu kumpulan besaran (variabel dan konstanta) yang dapat dirujuk melalui *indek*-nya, yang 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.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \text{... (P. 2.1)}$$

1. Penjumlahan dan pengurangan *matrix*.

Penjumlahan dan pengurangan *matrix* dapat dilakukan dengan mengoperasikan komponen *matrix* pada letak yang sama, atau dilambangkan dengan : $a_{ij} \pm b_{ij} = c_{ij}$

atau dalam representasi dekoratifnya :

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 2+5 \\ 1+7 & 0+5 & 0+0 \\ 1+2 & 2+1 & 2+1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 7 \\ 8 & 5 & 0 \\ 3 & 3 & 3 \end{bmatrix} \quad \text{... (P. 2.2)}$$

2. Perkalian Skalar.

Perkalian *matrix* dilakukan dengan mengalikan setiap elemen dari *matrix*, sebagai contoh :

$$2\begin{bmatrix}1 & 8 & -3\\ 4 & -2 & 5\end{bmatrix} = \begin{bmatrix}2\times 1 & 2\times 8 & 2\times -3\\ 2\times 4 & 2\times -2 & 2\times 5\end{bmatrix} = \begin{bmatrix}2 & 16 & -6\\ 8 & -4 & 10\end{bmatrix} \quad \text{... (P. 2.3)}$$

Perkalian 2 buah *matrix* digambarkan jika hanya jika jumlah kolom *matrix* kiri sama dengan jumlah baris *matrix* kanan, sebagai contoh :

$$\begin{bmatrix}3 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\ 4 & 2\end{bmatrix} = \begin{bmatrix}(3\times 1+1\times 4) & (3\times 1+1\times 2)\\ (1\times 1+0\times 4) & (1\times 1+0\times 2)\end{bmatrix} = \begin{bmatrix}7 & 5\\ 1 & 1\end{bmatrix} \quad \text{... (P. 2.4)}$$

3.2 Teori *Matrix* Grafik

1 *Adjacency Matrix* adalah sebuah *matrix* bujur sangkar yang mewakili suatu grafik, dengan a_{ij} 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.

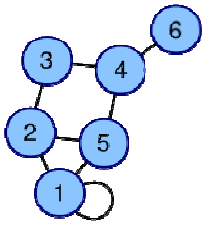
4 *Incidence Matrix* adalah sebuah matrix menggambarkan sebuah hubungan antara 2 kelas objek (biasanya puncak dan ujung dari konteks grafik).

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.

3.2.1 *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

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 2.1 ***Adjacency Matrix***

3.2.2 *Biadjacency Matrix*

Pada ilmu matematika dan ilmu komputer, *biadjacency matrix* pada grafik G dengan **n** *vertex* hitam dan m *vertex* putih adalah *matrix* **n x m** dimana a_{ij} adalah jumlah *edges* digabung *vertex* i putih dengan *vertex* j hitam. Pada kasus tertentu *biadjacency matrix* adalah (0,1)-*matrix*. *Adjacency Matrix* ***A*** pada sebuah grafik *bipartite* dengan *biadjacency matrix* B, di dapat dari :

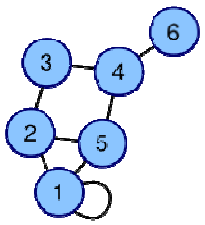
$$A = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}. \qquad \text{... (P. 2.5)}$$

Hubungan antara grafik *bipartite* dan *biadjacency matrix* dipelajari pada teori *grafik spectral*.

3.2.3 *Degree Matrix*

Pada ilmu matematika sub teori grafik, *Degree Matrix* adalah diagonal *matrix* yang mengandung informasi tentang derajat *vertex*.Berasal dari *G=(V,E)* dengan, *Degree Matrix* D untuk G adalah ***n x n*** *square matrix* sebagai :

Contoh :

Grafik *Vertex*	*Degree Matrix*
	$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Gambar 2.2 ***Degree Matrix***

3.2.4 *Incidence Matrix*

Pada ilmu matematika, *incidence matrix* adalah *matrix* yang menunjukkan hubungan antar dua kelas objek. Jika kelas utama X dan yang kedua Y, *matrix* mempunyai satu baris untuk elemen X dan satu kolom untuk elemen Y. Masukan dalam baris x dan kolom y adalah 1 jika x dan y terhubung dan 0 jika bukan.

3.2.5 *Laplacian Matrix*

Pada ilmu mathematika bidang teori grafik *Matrix Laplacian*, kadang-kadang disebut *admittance matrix* atau *Kirchhoff Matrix*, adalah suatu penyajian *matrix* suatu grafik. Bersama-sama dengan teori

Kirchhoff dapat digunakan untuk meng-kalkulasi banyaknya memutar pohon untuk grafik yang ditentukan.

Laplacian suatu grafik ***G*** digambarkan sebagai $\boldsymbol{L := D - A}$ dengan D *degree matrix* ***G*** dan A *adjacency* matrix ***G***. Lebih jelasnya diketahui grafik ***G*** dengan ***n*** *vertex*, matrix

$$l_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ adjacent } v_j \\ 0 & \text{otherwise} \end{cases} \quad \text{... (P. 2.6)}$$

3.2.6 *Seidel Adjacenty Matrix*

Pada ilmu matematika dan Teori *Matrix* Grafik, *Seidel Adjacency Matrix* pada grafik sederhana ***G*** (disebut juga *Seidel Matrix*) adalah *matrix* simetris dengan baris dan kolom untuk masing-masing *vertex*, mempunyai 0 dalam diagonal dan posisi yang bersesuaian dengan puncak $\boldsymbol{V}_i$ dan $\boldsymbol{V}_j$, -1 jika *vertices* bersebelahan dan +1 jika tidak bersebelahan. *Seidel Matrix* dikenalkan oleh *Van Lind* dan *Seidel* pada tahun 1966 dan secara aktif dimanfaatkan oleh *Seidel* dan *Coauthors*.

3.2.7 Pengujian

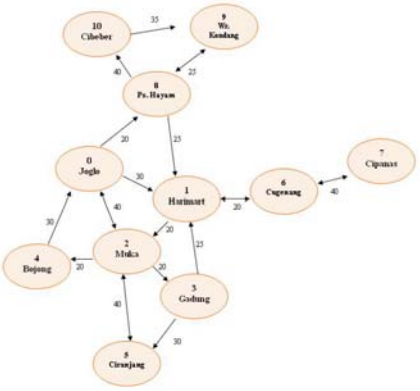
Dalam melakukan simulasi dari aplikasi *web based* analisis rute jalan yang telah ditempatkan didalam *web server*, diperlukannya adanya tahapan-tahapan dalam simulasi tersebut.

Untuk mengakses aplikasi ini pada komputer digunakan *Internet Explorer* sebagai *browser* kemudian memasukan *url " http://localhost/rute"* , dan apabila *web server* sudah aktif maka akan didapatkan tampilan seperti yang terlihat pada gambar 4.4. Tampilan yang tertera dalam gambar 4.4 adalah tampilan selamat datang, sehingga untuk melakukan perintah-perintah tertentu bisa digunakan *main menu* yang berada pada posisi sebelah kiri. Tampilan lebih jelas *main menu* dapat dilihat pada seperti pada gambar 4.5 di bawah.



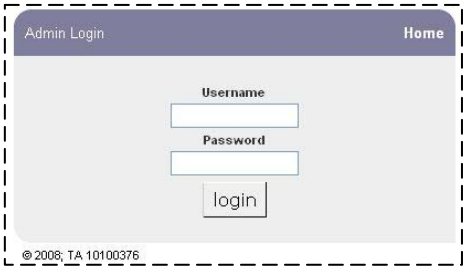
Gambar 4.3 *Main Menu* aplikasi rute jalan

Dari gambar 4.3 di atas dapat dijelaskan terdapat 4 buah menu utama yang dapat digunakan yaitu *Home, Rute Terpendek, Rute Alternatif* dan *Panel Rute*. Sebelum menggunakan menu *rute terpendek* dan *rute alternative*, terlebih dahulu data utama dan data pendukung untuk rute disi.



Gambar 4.4 Kasus Rute Jalan

Kasus di atas menjelaskan terdapat 11 buah tempat yang harus diisi ke dalam *Panel Tempat*, yaitu Joglo, Harimart, Muka, Gadung, Bojong dan Ciranjang, Cugenang, Cipanas, Pasir Hayam, Warung Kondang dan Cibeber. Maka dari menu utama pada gambar 4.6 klik menu *Panel Rute* untuk melihat seluruh menu *panel* yang ada, *Panel Rute* merupakan halaman khusus admin, sehingga ketika akan memasuki halaman-halaman tersebut akan diminta login terlebih dahulu, halaman login dapat dilihat pada gambar 4.5 .

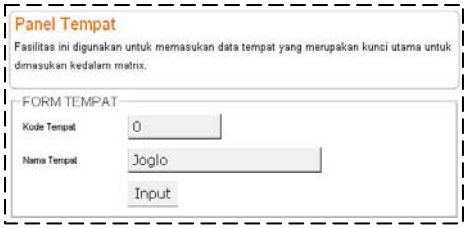


Gambar 4.5 *Form Login*

Apabil berhasil login makan akan muncul *Panel Tempat, Rute, Transportasi* dan *Jalur*. *Panel Menu* lebih jelas dapat dilihat pada gambar 4.6.



Gambar 4.6 *Panel Menu* aplikasi rute jalan



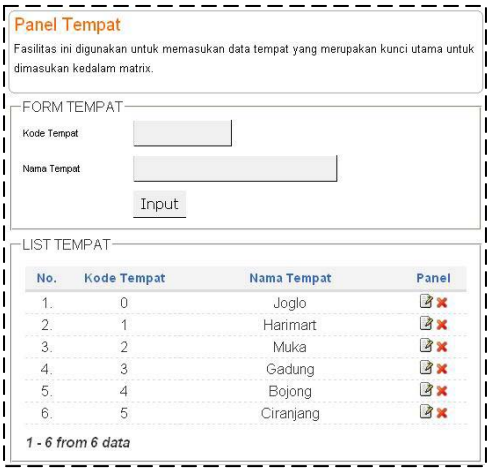
Gambar 4.7 *Input* Data Tempat

Isi data tempat pada *Form Tempat* seperti gambar 4.7 di atas kemudian klik tombol *Input* sehingga apabila berhasil *Form Tempat* akan memberikan pemberitahuan *Input Berhasil*.



Gambar 4.8 *Input* Data Tempat Sukses

Apabila ke-5 data tempat telah diinput, maka akan terlihat seperti gambar 4.9 dibawah.



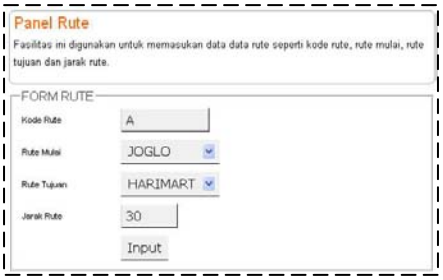
Gambar 4.9 *Form* dan *List* data Tempat

Salah satu data utama rute telah diinput, selanjutnya memasukan data rute sesuai dengan kasus pada gambar 4.4 diatas dengan meng-klik *Panel Rute*, data yang diperlukan untuk menjadi data rute sesuai dengan aplikasi yang dirancang yaitu kode rute, kode tempat, kode tempat tujuan dan jarak. Data ini fungsinya menghubungkan dari titik 1 ke titik lainnya dan dengan bantuan *Adjacency matrix* proses hubungan antar titik dapat didapat lebih mudah dan efesien hanya dengan membuat sebuah modul *Matrix.*

```
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 4.10 Ilustrasi *Matrix* pada *Array PHP*

Gambar 4.10 merupakan hasil olahan data 5 tempat yang diinput kemudian diproses melalui sebuah *sintax php* untuk menjadi *array*. Array diatas mempunyai nilai kosong dan belum ada hubungan dari satu titik ke titik lainnya.



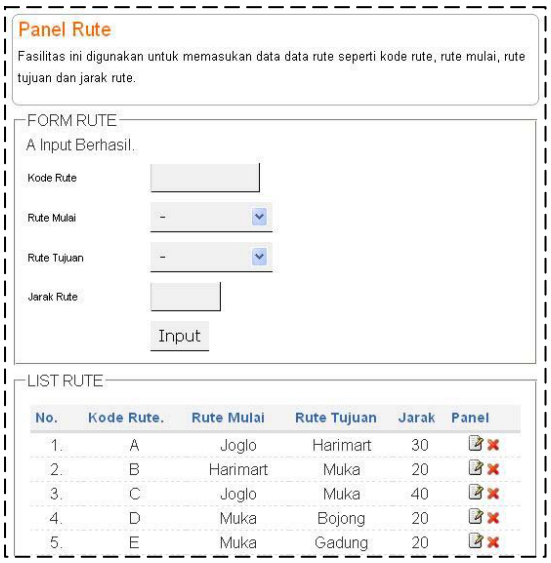
Gambar 4.11 *Input* data rute

Gambar 4.11 menjelaskan proses input data rute untuk membuat relasi antar titik sesuai kasus. Apabila semua data rute telah sukses input, ilustrasi *matrix* 4.10 berubah menjadi gambar 4.12 dibawah.

```
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 4.12 Ilustrasi Isi *Matrix Array PHP*

Gambar 4.12 tidak dibuat secara tampilan, yang ditampilkan merupakan daftar relasi antar titik melalui daftar rute yang telah diinput seperti gambar 4.13.



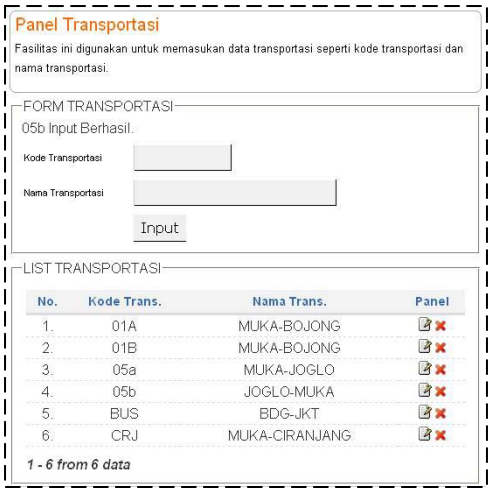
Gambar 4.13 *Form* dan *List* data Rute

Selanjutnya data yang harus diisi merupakan data pendukung rute yaitu data transportasi, data transportasi merupakan data yang direlasikan dengan data jalur, dimana untuk menuju suatu tujuan, transportasi merupakan data utama yang dipakai, berdasarkan perancangan data yang disimpan dalam data transportasi adalah kode transportasi dan nama transportasi. Untuk simulasi input data transportasi bisa diklik di menu transport dan akan muncul *Form Transportasi*.



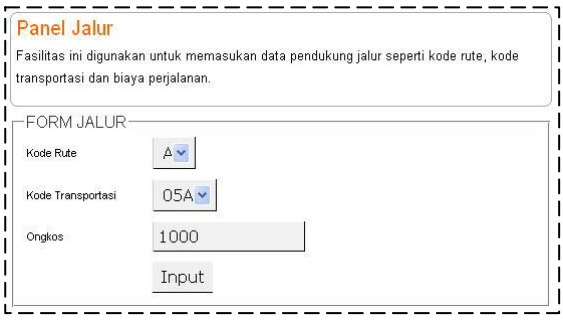
Gambar 4.14 *Input* Data Transportasi

Gambar 4.14 menjelaskan proses input untuk data transportasi 05A : MUKA-JOGLO, setelah form diisi selanjutnya klik tombol *Input* untuk mengeksekusi. Apabila semua data telah terinput, daftar lebih jelas dapat dilihat pada gambar 4.15.



Gambar 4.15 *Form* dan *List* dataTransportasi

Data pendukung lainnya rute adalah data jalur, data jalur mempunyai hubungan dengan data rute dan data transportasi, data rute berisi kode rute, kode transportasi dan ongkos. Untuk mengisi data jalur pada *Panel Menu* klik menu *Jalur* sehingga akan muncul seperti gambar 4.16 dibawah.

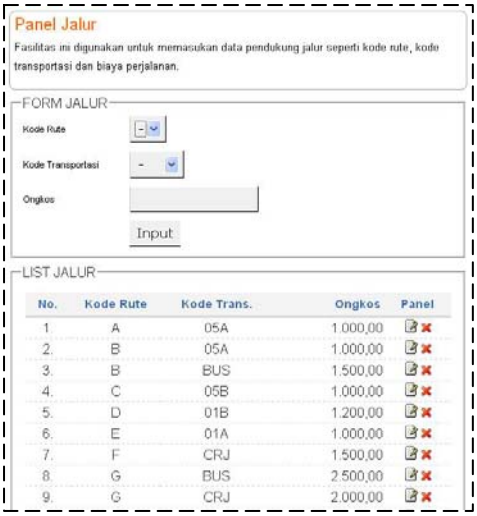


Gambar 4.16 *Input* data jalur

Gambar 4.16 menjelaskan proses untuk input data rute A bisa menggunakan kode transport 05A dan memerlukan ongkos Rp. 1000,-kedua data tadi (data transportasi dan data jalur) dapat menghasilkan data kemungkinan kendaraan yang digunakan berikut jarak dan ongkos yang harus dikeluarkan. Jadi secara konsep data kendaran tidak dapat dicari langsung tanpa melalui pencarian rute.

Untuk input data jalur data-data harus sesuai dengan dilapangan sehingga hasil pencarian dapat diakui dan bermanfaat.

Semua data yang telah diinput data dilihat didaftar yang telah disediakan aplikasi seperti pada gambar 4.17.



Gambar 4.17 *Form* dan *List* data Jalur

Gambar 4.17 menjelaskan daftar jalur yang diinput memiliki kode rute, kode transportasi dan ongkos. Sampai tahap ini semua data yang diperlukan untuk pemrosesan rute telah selesai antara lain data tempat, data rute, data transportasi dan data jalur. Pengisian semua data utama dan pendukung apabila benar dapat menghasilkan informasi yang bermaanfaat, sehingga aplikasi perancangan analisi rute jalan ini dapat dikatakan sukses.

Untuk selanjutnya akan dicoba simulasi rute alternative dan rute terpendek dari request yang diinput.

Apabila masih pada posisi *Panel Menu* maka, sebelumnya harus kembali ke *main menu* dengan mengklik menu *home* terlebih dahulu dan apabila telah pada posisi menu *main menu* klik menu *rute alternatife* untuk memulai simulasi pertama.



Gambar 4.18 Simulasi Rute Alternatif

Pada simulasi ini penulis mencoba memilih rute dari *joglo* menuju *muka*, dan hasil

pencarian menemukan 2 buah rute yang dapat dilalui :

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

Hasil pencarian rute diatas sama dengan yang tertera pada gambar 4.6 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 diatas adalah jalur tercepat dan termurah dapat menggunakan rute *alternatif* 2.

Simulasi selanjutnya adalah untuk mencari rute terpendek, penulis mencoba memasukan rute tujuan yang sama agar penelitian terhadap kasus tidak berbeda-beda.



Gambar 4.19 Simulasi Rute Terpendek

Gambar 4.19 diatas menghasil informasi rute menuju *muka* dari *joglo* mempunyai rute terpendek 40 KM dengan menggunakan transportasi 05B dan biaya ongkos sebesar Rp.1.000,-

Hasil tersebut sama persis nilainya ketika melakukan pencarian rute *alternatif*.

1. Simulasi penggunaan metode yang digunakan.

   Untuk simulasi misalkan dari data tempat yang telah diinput sebanyak 6 tempat seperti pada gambar 4.20

Kode Tempat	Nama Tempat
0	Joglo
1	Harimart
2	Muka
3	Gadung
4	Bojong
5	Ciranjang

Gambar 4.20 Data Tempat

Data tempat yang telah diinput diolah kembali ke dalam array dan menghasilkan ilustrasi seperti pada gambar 4.21

```
$m  = array(     // 0  1  2  3  4  5
            array( 0, 0, 0, 0, 0, 0), // 0
            array( 0, 0, 0, 0, 0, 0), // 1
            array( 0, 0, 0, 0, 0, 0), // 2
            array( 0, 0, 0, 0, 0, 0), // 3
            array( 0, 0, 0, 0, 0, 0), // 4
            array( 0, 0, 0, 0, 0, 0)  // 5
           );
```

Gambar 4.21 Ilustrasi Array Tempat

Untuk membaca ilustrasi pada gambar 4.21 dapat dilihat kolom sebanyak 6 dan baris sebanyak 6, dengan kata lain jika terdapat

kolom 6 buah pasti akan terdapat pula baris sebanyak 6 buah. Semua nilai dalam ilustrasi array diatas adalah 0 (nol) , cara pembacaan dimulai dari baris menuju kolom, misalkan baris 1 menuju kolom 2 hasilnya 0. *Adjacency Matrix* merupakan teori yang menghubungkan dari titik satu ke titik lainnya atau dari ilustrasi *array* dinyatakan menghubungkan baris ke kolom. Untuk mengisikan nilai pada titik ke titik perlu diinput kembali data relasi antar titik, data rute dapat dilihat pada gambar 4.22.

Kode Rute.	Rute Mulai	Rute Tujuan	Jarak (KM)
A	Joglo	Harimart	30
B	Harimart	Muka	20
C	Joglo	Muka	40
D	Muka	Bojong	20
E	Muka	Gadung	20
F	Gadung	Ciranjang	30
G	Muka	Ciranjang	40
H	Muka	Joglo	40
I	Ciranjang	Muka	40
J	Bojong	Joglo	30

Gambar 4.22 Data Rute

Dari gambar 4.22 yang dicocokan dengan gambar 4.23 pada kode rute A menghubungkan titik 0=Joglo ke titik 1=Harimart sebanyak 30.

```
$m  = array(    // 0   1   2  3  4  5
            array( 0,30, 40, 0, 0, 0), // 0
            array( 0, 0, 20, 0, 0, 0), // 1
            array( 40, 0, 0, 20,20,40), // 2
            array( 0, 0, 0,  0, 0,30), // 3
            array( 30, 0, 0,  0, 0, 0), // 4
            array( 0, 0,40,  0, 0, 0)  // 5
          );
```

Gambar 4.23 Ilustrasi Isi Array Tempat

Setelah array tempat terisi langkah selanjutnya adalah mengisi data jalur dan data transportasi, dimana data transportasi mempunyai hubungan banyak ke banyak dengan data jalur dan data jalur mempunyai hubungan banyak ke banyak pada data rute.

Kode Trans.	Nama Trans.
01A	MUKA-BOJONG
01B	MUKA-BOJONG
03	PSHAYAM-HARIMART
05A	MUKA-JOGLO
05B	JOGLO-MUKA
06	BOJONG-JOGLO
BUS	BDG-JKT
CBR	JOGLO-CIBEBER
CPS	HARIMART-CIPANAS
CRJ	CIRANJANG

Gambar 4.24 Data Transportasi

Kode Rute	Kode Trans.	Ongkos
A	05A	1.000,00
B	05A	1.000,00
B	BUS	1.500,00
C	05B	1.000,00
D	01B	1.200,00
E	01A	1.000,00
F	CRJ	1.500,00
G	CRJ	2.000,00
G	BUS	2.500,00
H	05A	1.000,00

Gambar 4.25 Data Jalur

Dan ketika dilakukan pencarian rute terpendek dan rute alternatif dengan bantuan data transportasi dan data jalur akan menghasilkan tampilan sebagai berikut :



Gambar 4.26 Hasil Pencarian Rute Joglo ke Harimart

1. Kesimpulan dan Saran

Dari hasil penelitian didapatkan kesimpulan dan saran sebagai berikut.

1.1 Kesimpulan

Hasil kesimpulan yang didapat, antara lain :

1. Dalam mengelola rute jalan yang banyak dan komplek , konfigurasi rute jalan pada aplikasi komputer dapat membantu seorang menemukan rute jalan secara cepat. Metode *Matrix* merupakan solusi yang tepat untuk menangani konfigurasi rute jalan tersebut. Data utama dan data pendukung rute merupakan faktor olahan yang harus dimasukan 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. Konfigurasi rute jalan yang digunakan harus menggunakan infrastruktur yang dapat diakses dengan cepat bila perlu secara online 24 jam, infrastruktur tersebut merupakan basis web. Bahasa web PHP terbukti memiliki kemampuan paling cepat, paling aman dan murah dibandingkan dengan bahasa web lainnya.
3. Kemudahan dalam mengatur konfigurasi rute jalan perlu diperhatian . Fungsi dari aplikasi *web based* yang dirancang telah mempermudah konfigurasi rute jalan. Aplikasi yang dirancang dapat membantu administrasi rute dalam memonitoring dan dapat menciptakan informasi rute jalan yang tepat sesuai dengan kenyataanya.

3.1 Saran

Saran yang diperoleh diantaranya :

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 komplek.
3. Pada aplikasi yang dirancang perlu ditambahkan fungsi – fungsi yang dapat membantu administrasi rute memonitoring lalu lintas jalan.

4. 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_%28mathematics%29
6. http://id.wikipedia.org/wiki/Matriks_%28matematika%29
7. http://www.brpreiss.com/books/opus4/html/page533.html
8. http://www.cs.usask.ca/resources/tutorials/csconcepts/1999_8/tutorial/beginner/matrices/matrix.html
9. Roger S Pressman (2002),“ Rekayasa Perangkat Lunak Pendekatan Praktisi “, Andi, Yogyakarta
10. Yahya Kurniawan (2002), “ Aplikasi Web Database dengan PHP dan MySQL”, Elex Media Komputindo,Jakarta
1.