Penentuan Jarak Terpendek Rute Transmusi dengan Algoritma *Floyd-Warshall*

Y. Rudi Kriswanto[1], R. Kristoforus Jawa Bendi[2], Arif Aliyanto[3]

[1]Teknik Informatika, Sekolah Tinggi Teknik Musi, Palembang 30113
Email:[1]johanes27capatar@gmail.com,[2]kristojb@sttmusi.ac.id

[3]Sistem Informasi, Sekolah Tinggi Teknik Musi, Palembang 30113
Email:aliyanto_arif@sttmusi.ac.id

ABSTRAK

Transmusi merupakan sarana transportasi pulik di kota Palembang. Sepanjang rute transmusi tersedia halte-halte keberangkatan dan kedatangan.Kebanyakan penumpang kesulitan ketika harus menentukan jarak terdekat dari satu tempat ke tempat lainnya. Penelitian ini bertujuan mengembangkan perangkat lunak aplikasi untuk menentukan jarak terdekat yang dapat dilalui penumpang.
Penelitian ini menggunakan algoritma Floyd-warshall untuk menghitung jarak terdekat antar dua titik. Model proses pengembangan perangkat lunak yang digunakan adalah model waterfall. Perangkat lunak diaplikasikan dengan PHP , CSS , Javascript dan SQL Server 2008.
Hasil penelitian menunjukkan bahwa perangkat lunak yang dibangun dapat menjalankan algoritma Floyd-warshall dengan baik. Dengan demikian, aplikasi ini dapat igunakan untuk menentukan jarak terdekat yang dapat dilalui penumpang transmusi.

Kata kunci:*, tranportasi, algoritma jarak terpendek, Floyd-Warshall.*

1. Pendahuluan

Pencarian rute terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian rute terpendek ini telah diterapkan di berbagai bidang untuk mengoptimasi kinerja suatu sistem baik untuk meminimalkan biaya ataupun mempercepat jalannya suatu proses. Salah satu aplikasi pencarian rute terpendek yang paling menarik untuk dibahas adalah pada masalah transportasi [7].

Pencarian rute terpendek termasuk dalam salah satu persoalan dalam teori graf yang berarti meminimalisasi bobot suatu lintasan dalam graf [8]. Algoritma *floyd-Warshall* dapat menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik. Dengan kata lain pada saat perhitungan rute optimum yang akan dilalui terlebih dahulu menghitung semua kemungkinan rute yang akan dilalui kemudian mencari rute optimum dengan cara membandingkan setiap pasangan rute.

Di Kota Palembang terdapat transportasi umum milik pemerintah daerah yaitu *Bus Rapid Transit* (BRT) atau yang sering disebut dengan transmusi. Rute perjalanan transmusi dibedakan berdasarkan atas 6 koridor yang tersebar di Kota Palembang. Pada Sepanjang koridor terdapat halte-halte yang mempunyai nama sesuai dengan alamat tempat tersebut.

Berdasarkan observasi, didapatkan bahwa lebih dari 34 persen pengguna transmusi ternyata pernah mengalami kesalahan dalam memilih jalur transmusi. Akibatnya,banyak waktu mereka yang terbuang. Selain itu juga didapatkan hasil bahwa lebih dari 68 persen pengguna transmusi tidak mampu menentukan titik terdekat antar halte sehingga mereka tidak dapat menghemat waktu perjalanan mereka.

Untuk mengatasi permasalahan itu maka diperlukan adanya suatu aplikasi yang dapat membantu menentukan jalur terpendek pada rute transmusi sehingga dengan aplikasi seperti ini penggunaan waktu menjadi lebih efekfif dan permasalahan penumpang atau calon penumpang dapat diselesaikan.

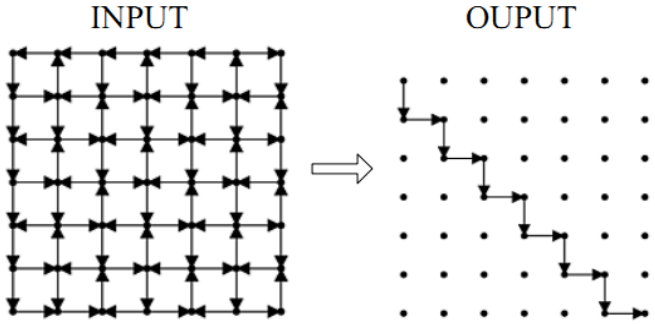
2. Tinjauan Pustaka

Transportasi didefinisikan sebagai perpindahan dari suatu tempat ke tempat lain dengan menggunakan alat pengangkutan, baik yang digerakkan oleh tenaga manusia, hewan atau mesin [12]. Konsep transportasi didasarkan pada adanya perjalanan (*trip*) antara asal (*origin*) dan tujuan (*destination*). Perjalanan adalah pergerakan orang dan barang antara dua tempat kegiatan yang terpisah untuk melakukan kegiatan perorangan atau kelompok dalam masyarakat. Perjalanan dilakukan melalui suatu lintasan

tertentu yang menghubungkan asal dan tujuan, menggunakan alat angkut atau kendaraan dengan kecepatan tertentu. Menurut [3] terdapat lima unsur pokok transportasi. Kelima unsur tersebut yaitu: a)manusia, yang membutuhkan transportasi, b) barang, yang diperlukan manusia, c)kendaraan, sebagai sarana transportasi,d)jalan, sebagai prasarana transportasi,e)organisasi, sebagai pengelola transportasi.Halte merupakan salah satu fasilitas transportasi untuk menaikkan atau menurunkan penumpang yang dilengkapi dengan bangunan [1]. Halte disediakan sebagai pendukung dalam mewujudkan sistem transportasi yang efektif dan efisien. Halte diperlukan keberadaannya di sepanjang rute angkutan umum dan angkutan umum harus melalui tempat yang telah ditetapkan untuk menaikkan dan menurunkan penumpang agar perpindahan penumpang lebih mudah dan gangguan terhadap lalu lintas dapat diminimalkan.

Menurut Siang [10] graf adalah kumpulan simpul *(titik)* yang dihubungkan satu sama lain melalui sisi/busur *(lintasan).* Suatu graf G terdiri dari 2 himpunan yang berhingga, yaitu himpunan titik titik kosong *(symbol V(G))* dan himpunan garis garis *(symbol E(G)).* Setiap garis berhubungan dengan satu atau dua titik. Titik titik tersebut dinamakan titik ujung. Garis yang hanya berhubungan dengan satu titik ujung disebut *loop.* Dua garis berbeda yang menghubungkan titik yang sama disebut garis paralel. Dua titik dikatakan berhubungan (*adjacent)* jika ada garis yang menghubungkan keduanya. Titik yang tidak memiliki garis yang berhubungan dengannya disebut titik tersaing *(isolating point).* Graf yang tidak memiliki titik sehingga tidak memiliki garis disebut graf kosong.

Dalam teori graf, persoalan lintasan terpendek *(the shortest path problem)* merupakan suatu persoalan untuk mencari lintasan antara dua buah titik pada graf berbobot yang memiliki gabungan nilai jumlah bobot pada sisi graf yang dilalui dengan jumlah yang paling minimum. Gambar 1 menunjukkan contoh penyelesaian *shortest path problem*[6].



Gambar 1*Shortest Path Problem*

Algoritma *floyd-warshall* adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah 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 [5].

Dalam usaha untuk mencari lintasan terpendek, algoritma *floyd-warshall* memulai iterasi dari titik awalnya kemudian memperpanjang lintasan dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminimum mungkin.Misalkan W_0 adalah martiks hubung graf berarah berlabel mula-mula. *W** adalah matriks hubung minimal dengan W_{ij}*= lintasanterpendek dari titik v_i ke v_j. *Algoritma floyd-warshall* untuk mencari lintasan terpendek adalah sebagai berikut [10].

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 = matriks
W_0= Martik hubung graf mula-mula
k = iterasi 1 sampai ke-n
i = titik awal pada vi
j = titik akhir pada vj
W=hasil matriks setelah perbandingan*

Dalam iterasinya untuk mencari lintasan terpendek, algoritma *floyd-warshall* membentuk *n* matriks sesuai dengan *iterasi-k.* Itu menyebabkan waktu prosesnya lambat, terutama untuk *n* yang besar. Meskipun waktu prosesnya bukanlah yang tercepat, algoritma *floyd-warshall* sering dipergunakan untuk menghitung lintasan terpendek karena kesederhanaan algoritmanya.

Algoritma ini menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik. Dengan kata lain pada saat perhitungan rute optimum yang akan dilalui terlebih dahulu. Algoritma *floyd-warshall* bekerja berdasarkan formulasi *dynamic programming*. Setiap langkahnya akan memeriksa lintasan antara v_i dan $v_j$apakah bisa lebih pendek jika melalui v_i-v_k dan v_k-v_j.Berikut ini pseudocode algoritma *floyd-warshall*[9].

```
function fw(int[1..n,1..n] graph)
{
 // Inisialisasi
var int[1..n,1..n] jarak := graph
 var int[1..n,1..n] sebelum
for i from 1 to n
for j from 1 to n
if jarak[i,j] < Tak-hingga
            sebelum[i,j] := i
 // Perulangan utama pada algoritma
for k from 1 to n
 for i from 1 to n
for j from 1 to n
if jarak[i,j] > jarak[i,k] + jarak[k,j]
jarak[i,j] = jarak[i,k] + jarak[k,j] sebelum[i,j] = sebelum[k,j]
 return jarak
  }
```

Penelitian-penelitian yang menggunakan algoritma ini telah banyak dilakukan. Hulliyah dan Fauzi [4] dalam penelitiannya membahas mengenai pembangunan sistem untuk menemukan jalur terpendek dengan menyertakan faktor kecepatan dan waktu tempuh perjalanan pada jalan raya wilayah blok M dan kota. Sistem ini dibangun dengan menggunakan bahasa pemrograman *PHP 5.2.2, My SQL 5.0.41* sebagai basis datanya dan x*ampp 1.3.2* sebagai web servernya. Dari penelitian ini didapatkan hasil bahwa algoritma *dijkstraa* mampu memberikan solusi untuk menemukan lintasan tercepat dan terpendek pada jalan blok M dan jalan kota. Jika penelitian ini menggunakan algoritma *dijkstraa*, maka penelitian yang dilakukan menggunakan algoritma *floyd-warshall.*

Fanani *et al*[3] dalam penelitiannya membahas mengenai pembangunan sistem informasi website pencarian rute terpendek di gedung kampus. Sistem ini dibangun dengan menggunakan bahasa pemrograman *PHP dan MySQL* serta *adobe flash* sebagai visualisasi rute. Berdasarkan hasil penelitian didapatkan bahwa tingkat akurasi algoritma *floyd-warshall* menunjukkan akurasi 100%. Persamaan penelitian ini yaitu pada penggunaan algoritma *floyd-warshall*. Perbedaan dalam kedua penelitian ini adalah jika objek pada penelitian ini adalah letak gedung Universitas Brawijaya, maka objek pada sistem yang akan dibangun adalah transportasi umum.

Penelitian lain membahas mengenai pembangunan sistem informasi geografis yang dijadikan sebagai alat bantu atau sebagai media wisatawan untuk mencari suatu lokasi atau penuntun arah ke suatu lokasi yaitu dengan mencari rute optimum lokasi objek wisata tertentu [9]. Sistem ini dibangun dengan menggunakan bahasa pemrograman *Visual Basic 6*. Hasil dari penelitian ini didapatkan bahwa algoritma *floyd-warshall* dapat digunakan dalam mencari rute optimum dengan pemrograman visual. Persamaan penelitian ini dengan penelitian yang akan dilakukan adalah sama sama mencari rute terdekat dan persamaan *algoritmafloyd-warshall*, namun perbedaan penelitian yang dilakukan dalam penelitian ini adalah fungsinya Jika penelitian ini menitikberatkan pada sistem informasi geografisnya maka penelitian yang akan dilakukan menitikberatkan pada aplikasi pencarian rute terpendek.

Sukrisno dan Rahman [13] membahas mengenai pengembangan algoritma *floyd-warshall* dalam penentuan *feasible route* dalam proses evakuasi bangunan bertingkat. Persamaannya adalah sama-sama menggunakan algoritma *floyd-warshall.* Perbedaannya adalah obyek penelitiannya. Jika objek penelitian ini adalah gedung bertingkat maka objek yang akan dilakukan penelitian adalah masalah transportasi umum. Dalam penelitian ini dikembangkan untuk merancang mekanisme *dynamic exit sign system* dalam sebuah blok sebagai sebuah sistem yang terotomasi dan membangun *prototype dynamic exit sign* dalam sebuah model bangunan bertingkat. Dari hasil penelitian ini didapatkan hasil bahwa algoritma *floyd-warshall* memiliki performansi yang stabil dan proses kalkulasi yang cepat dalam menentukan *shortest path problem.* Dari hasil tersebut maka akan dilakukan penelitian yang akan dilakukan dalam kebutuhan transportasi.

Dewi [2] dalam penelitiannya membahas mengenai pembangunan sistem yang berupa sistem informasi geografis yang berbentuk web yaitu dibangun menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai basis datanya. Dari penelitian ini didapatkan hasil bahwa algoritma dijsktra cukup baik dalam pencarian rute terpendek suatu tempat wisata di Bali. Persamaan penelitian ini adalah terletak pada persamaan fungsinya yaitu mencari rute terdekat atau terpendek namun perbedaanya sistem yang sudah dibangun menggunakan algoritma *dijkstraa* sedangkan sistem yang akan dibangun yaitu menggunakan algoritma *floyd warshall* . Perbedaan lainnya adalah jika pada penelitian ini sistem digunakan untuk wisatawan, maka sistem yang akan dibangun dalam penelitian ini digunakan oleh penumpang transmusi.

Pradhana [6] di dalam penelitiannya membahas mengenai studi dan implementasi persoalan lintasan terpendek suatu graf. Dari hasil penelitian ini didapatkan bahwa algoritma *djisktraa* hanya dapat digunakan pada graf yang berbobot positif sedangkan algoritma *bellman ford* dapat digunakan graf dengan bobot positif maupun negatif. Jika pada penelitian ini algoritma yang digunakan adalah algoritma *dijkstraa* dan algoritma *bellman-ford*, maka dalam penelitian yang akan dilakukan yaitu membangun sistem menggunakan *algoritma floyd-warshall.* Selain itu jika pada penelitian ini mengimplementasikan sistem dengan permasalahan matematika, maka penelitian yang akan dilakukan mengimplementasikan pada persoalan transportasi umum.

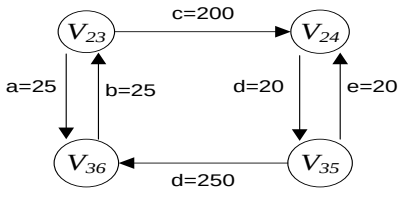
Rachmah [8] dalam penelitiannya membahas tentang pencarian lintasan terpendek graf dengan menyelesaikan persoalaan graf pada matematika menggunakan algoritma *dijsktraa*. Hasil dari penelitian ini yaitu bahwa algoritma *dijkstraa* dapat digunakan untuk mencari lintasan terpendek graf secara efisien karena tidak membutuhkan banyak waktu. Dari hasil penelitian tersebut maka penelitian yang akan dilakukan yaitu membangun sistem untuk diimplementasikan pada rute transmusi dengan menggunakan algoritma *floyd warshall*.

Suherman *et al*[11] menyimulasikan rute terpendek ke setiap host tujuan dalam suatu topologi jaringan dengan menggunakan algoritma *dijsktraa.* Dari penelitian ini didapatkan hasil bahwa algoritma *dijsktraa* mampu mensimulasikan penentuan rute terbaik tiap titik pada topologi jaringan. Jika dalam penelitian ini mengunakan algoritma *dijkstraa* yang diterapkan pada topologi jaringan, maka penelitian yang akan dilakukan menggunakan algoritma *floyd-warshall* pada transportasi umum. Tabel 2.4 menunjukkan perbandingan studi literatur.

3. Metode Penelitian

Analisa Graf

Masukkan algoritma *floyd-warshall* adalah matriks hubung graf yang mempunyai arah dan bobot. Dalam kasus ini halte, jarak antar halte, dan lintasan transmusi merupakan komponen yang disebut graf berarah dan berbobot. Halte disebut dengan titik, sedangkan jarak antar halte disebut dengan lintasan. Jumlah halte pada koridor 1 dan koridor 2 sebanyak 115 halte. Halte-halte tesebut mempunyai nama sesuai dengan tempat pemberhentian. Dari halte satu ke halte yang lain terdapat lintasan yang mempunyai jarak tertentu. Lintasan ini merupakan jalan raya yang dilewati oleh bus transmusi. Didalam kasus ini juga ditambahkan lintasan yang tidak dilalui bus transmusi. Lintasan tersebut ditambahkan sebagai penghubung antar halte yang berseberangan. Lintasan ini digunakan sebagai lintasan penumpang yang akan melakukan penyeberangan. Gambar 2 menunjukkan sampel beberapa halte, jarak antar halte, dan lintasan yang diimplementasikan kedalam sebuah graf berbobot dan berarah.



Gambar 2 Sampel Halte dan Lintasan

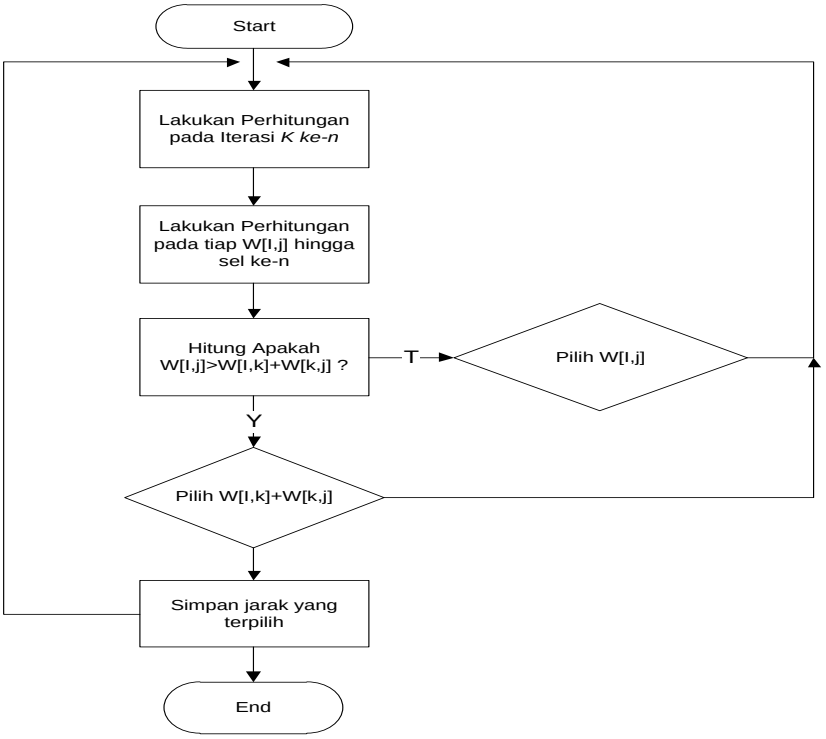
Keterangan **:** Halte: V_{23}, V_{24}, V_{35}, V_{36}
Lintasan transmusi : c, d
Lintasan penumpang: a, b, d, e

Analisis Kebutuhan Fungsional

Kebutuhan fungsional dari sistem ini meliputi;
a. sistem ini akan menerima input awal yaitu terdiri dari halte asal dan halte tujuan
b. sistem harus mampu memberikan informasi bagi *user* seperti; informasi mengenai nama halte yang dilalui, jarak antar halte, dan total jarak terdekat yang dilalui *user*.
c. sistem memberikan informasi mengenai rute terdekat transmusi sesuai dengan asal dan tujuan yang dipilih oleh penumpang.

Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah deskripsi dari fitur-fitur, karakteristik, dan batasan-batasan yang mendefinisikan sistem. Kebutuhan non-fungsional dari sistem ini meliputi;
a. waktu respons; sistem yang dibangun dapat digunakan secara cepat.
b. tampilan; sistem yang dibangun harus bersifat *user friendly* bagi pengguna dengan tampilan dan keterangan yang jelas dan mudah dipahami.



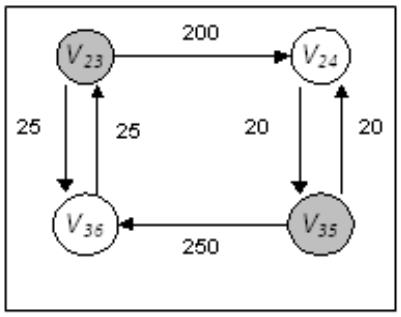
Gambar 3 algoritma *floyd-warshall*

Proses Penentuan Nilai Minimum Algoritma *Floyd-Warshall*

Proses penentuan nilai minimum algoritma *floyd-warshall* dapat dituliskan sebagai berikut:
a. Pada iterasi ke-1, setiap sel matriks dilakukan pengecekan apakah jarak antar dua titik mula mula lebih besar dari penjumlahan antar jarak titik asal ke titik tujuan (titik tujuan=iterasi ke-1) dengan jarak titik asal (titik asal=iterasi ke-1) ke titik tujuan. Dengan kata lain apakah $W[i,j] > W[i,k] + W[k,j]$.
b. Jika iya maka jarak antar dua titik mula mula diganti dengan penjumlahan antar jarak titik asal ke titik tujuan (titik tujuan=iterasi ke-1) dengan jarak titik asal (titik asal=iterasi ke-1) ke titik tujuan $(W[i,k] + W[k,j])$.
c. Jika tidak, maka jarak yang digunakan yaitu jarak antar dua titik mula mula ($W[i,j]$).
d. Proses iterasi dilakukan hingga pada iterasi terakhir (jumlah iterasi=jumlah total titik). Gambar 3 menunjukkan *flowchart* alur penentuan nilai minimum pada algoritma *floyd-warshall.*

Penerapan Algoritma *Floyd-Warshall*

Pada penerapan algoritma *floyd-warshall*, koridor 1 merupakan koridor yang digunakan sebagai sampel dan diberikan contoh bahwa V_{23} merupakan Halte BNI, V_{24} merupakan Halte Gereja St.Yosep, V_{35}merupakan Halte RS Charitas, dan V_{36} merupakan Halte Hotel Djayakarta. Dalam kasus ini akan dicari rute terdekat dari titik asal V_{23} (Halte BNI*)* dan titik tujuan V_{35} (RS Charitas). Gambar 4menunjukkan sampel halte pada koridor 1.



Gambar 4 sampel halte pada koridor 1.

Keempat halte tersebut merupakan titik yang digunakan sebagai sampel dalam implementasi algoritma *floyd-warshall*. Keempat halte tersebut masing-masing mempunyai jarak antar halte. Tabel 1 menunjukkan jarak antara halte asal dengan halte tujuan.

Tabel 1Jarak Antar Halte

Asal	Tujuan	Jarak (m)
V_{23}	V_{23}	∞
V_{23}	V_{24}	200
V_{23}	V_{35}	∞
V_{23}	V_{36}	25
V_{24}	V_{23}	∞
V_{24}	V_{24}	∞
V_{24}	V_{35}	20
V_{24}	V_{36}	∞
V_{35}	V_{23}	∞
V_{35}	V_{24}	20
V_{35}	V_{35}	∞
V_{35}	V_{36}	250
V_{36}	V_{23}	25
V_{36}	V_{24}	∞
V_{36}	V_{35}	∞
V_{36}	V_{36}	∞

Dibawah ini merupakan proses penyelesaian untuk menemukan jarak terdekat dari titik asal V_{23} (Halte BNI*)* dan titik tujuan V_{35} (RS Charitas).

a. Menentukan semua kemungkinan rute yang dapat dilalui dari titik asal V_{23} menuju titik tujuan V_{35}.
b. Untuk menentukan semua kemungkinan rute yang dapat dilalui tersebut dapat dilihat dalam matriks pada iterasi ke-0 dan membentuk matriks $W_{0.}$ Gambar 4menunjukkanmatriks $W_{0.}$

Pada rute 1 dan rute 2, V_{24} dan V_{36} merupakan titik terakhir pada iterasi ke-0. Selanjutnya dilakukan perhitungan iterasi ke-1. Dalam perhitungan iterasi ke-1, W_0 dipakai sebagai acuhan sehingga menghasilkan iterasi ke-1, demikian seterusnya sehingga sampai pada perhitungan iterasi ke-4 adalah sebagai berikut.

$$W_0 = \begin{array}{c} \\ V_{23} \\ V_{24} \\ V_{35} \\ V_{36} \end{array} \begin{array}{c} \begin{array}{cccc} V_{23} & V_{24} & V_{35} & V_{36} \end{array} \\ \begin{pmatrix} \infty & 200 & \infty & 25 \\ \infty & \infty & 20 & \infty \\ \infty & 20 & \infty & 250 \\ 25 & \infty & \infty & \infty \end{pmatrix} \end{array} \quad W_1 = \begin{array}{c} \\ V_{23} \\ V_{24} \\ V_{35} \\ V_{36} \end{array} \begin{array}{c} \begin{array}{cccc} V_{23} & V_{24} & V_{35} & V_{36} \end{array} \\ \begin{pmatrix} \infty & 200 & \infty & 25 \\ \infty & \infty & 20 & \infty \\ \infty & 20 & \infty & 250 \\ 25 & 225 & \infty & 50 \end{pmatrix} \end{array}$$

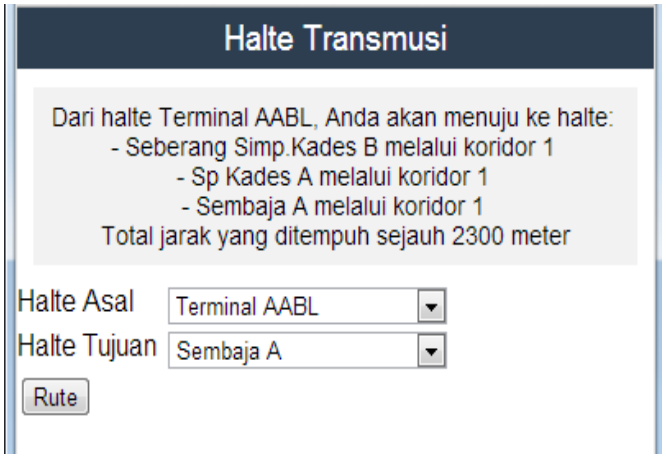
$$W_2 = \begin{array}{c} \\ V_{23} \\ V_{24} \\ V_{35} \\ V_{36} \end{array} \begin{array}{c} \begin{array}{cccc} V_{23} & V_{24} & V_{35} & V_{36} \end{array} \\ \begin{pmatrix} \infty & 200 & 220 & 25 \\ \infty & \infty & 20 & \infty \\ \infty & 20 & 40 & 250 \\ 25 & 225 & \infty & 50 \end{pmatrix} \end{array}$$

$$W_3 = \begin{array}{c} \\ V_{23} \\ V_{24} \\ V_{35} \\ V_{36} \end{array} \begin{array}{c} \begin{array}{cccc} V_{23} & V_{24} & V_{35} & V_{36} \end{array} \\ \begin{pmatrix} \infty & 200 & 220 & 25 \\ \infty & 40 & 20 & 270 \\ \infty & 20 & 40 & 250 \\ 25 & 225 & 245 & 50 \end{pmatrix} \end{array} \quad W_4 = \begin{array}{c} \\ V_{23} \\ V_{24} \\ V_{35} \\ V_{36} \end{array} \begin{array}{c} \begin{array}{cccc} V_{23} & V_{24} & V_{35} & V_{36} \end{array} \\ \begin{pmatrix} 50 & 200 & 220 & 25 \\ 259 & 40 & 20 & 270 \\ 275 & 20 & 40 & 250 \\ 25 & 225 & 245 & 50 \end{pmatrix} \end{array}$$

Dalam penyelesaian kasus ini diperoleh informasi bahwa rute yang mempunyai jarak terdekat dari (V_{23})Halte BNI menuju (V_{35})Halte RS Charitas yaitu melalui (V_{23})Halte Rs Charitas degan menggunakan Bus K1 kemudian melalui (V_{24}) Halte Gereja St.Yosep, selanjutnya menuju (V_{35}) Halte RS Charitas dengan menyeberang, dan total jarak tempuh adalah 220 meter.

4. Hasil dan Pembahasan

Hasil utama dari sistem ini ditunjukkan pada *form*rute. Tampilan form ini merupakan tampilan informasi rute transmusi dengan jarak terdekat. Pada *form* ini *u*ser dapat memperoleh informasi mengenai halte yang harus dilalui. Gambar 5 menunjukkan *form rute*



Gambar 5 menunjukkan *form rute*

Didalam penelitian ini ditemukan beberapa kendala. Kendala tersebut adalah sebagai berikut.

a. Data yang diperoleh dari Perusahaan BRT transmusi adalah data operasional bus transmusi seperti; jumlah koridor, nama-nama halte, denah koridor dsb. Dalam hal ini pada perusahaan tidak mempunyai data jarak antar halte, sehingga diperlukan pengumpulan data jarak halte dengan melakukan pengukuran jarak antar halte sendiri.
b. Dalam kegiatan pengukuran jarak antar halte dilakukan dengan menggunakan motor Supra Fit-X yaitu dengan cara mengukur menggunakan spedometer motor tersebut, sehingga diperlukan waktu yang cukup lama.
c. Dalam kegiatan penelitian juga terdapat kesulitan dalam mencari letak halte sehingga mengalami tersesat.
   Selain itu didapatkan beberapa hasil dari penelitian ini. Hasil penelitian tersebut adalah sebagai berikut.
   1) Sistem yang dihasilkan dapat memberikan informasi rute halte terdekat kepada *user* secara cepat dan tepat tanpa memerlukan waktu yang cukup lama.

2) Terdapat beberapa *form* pada *admin* ketika diakses sedikit memerlukan waktu yang lebih lama. Misalnya pada *form* edit rute aksesnya sedikit lebih lamadikarenakan data yang ditampung cukup banyak.

5. Simpulan

Berdasarkan hasil penelitian yang telah dilakukan didapatkan kesimpulan bahwa. Algoritma *floyd-warshall* dapat diimplementasikan untuk memperkirakan jarak terdekat pada rute transmusi Palembang.

Adapun saran yang dapat disampaikan pada penelitian yang akan datang dikemudian hari yaitu sebagai berikut.

a. Sebaiknya objek penelitian tidak hanya mencakup dua koridor saja melainkan semua koridor pada rute transmusi palembang ini.
b. tampilan *visualisasi* rute sistem ini sebaiknya dapat ditambahkan untuk meningkatkan kemudahan *user* dalam mencari rute transmusi.
c. Pada penelitian selanjutnya sebaiknya dikembangkan sistem tidak hanya menentukan jarak terdekat melainkan dikembangkan sistem untuk menentukan waktu tercepat atau menentukan biaya terhemat.

Daftar Pustaka

[1] Daud, Jeluddin., 2005, Studi Efektifitas Penggunaan Halte di Kota Medan, *Jurnal Sistem Teknik Industri*, No 3, Vol. 6.

[2] Dewi, Luh Joni Erawati., 2010, Pencarian Rute Terpendek Tempat Wisata di Bali Menggunakan Algoritma Dijsktra, *Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010)*, 19 Juni 2010.

[3] Fanani, Lutffi; J, Eriq M Adams; Wicaksono, Satrio A., 2012, Rancang Bangun Aplikasi Web Pencarian Rute Terpendek Antar Gedung di Kampus Menggunakan Algoritma Floyd-warshall, *Jurnal Basic Science And Techonology*, 1(3),7-11,2012 ISSN : 2089-8185, Malang.

[4] Hulliyah, Khodijah; Fauzi, Imron., 2011, Implementasi Algoritma Djikstra untuk Mendapatkan Jalur Tercepat dan Jalur Terpendek,Seminar Nasional Teknologi Informasi, Komunikasi & Aplikasinya, Vol.1, No. 1, 2011, 11 Nov 2011.

[5] Novandi, Raden Aprian Diaz., 2007, Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam penentuan Lintasan Terpendek (Single Pair Shortest Path), *Makalah IF2251 Strategi Algoritmik*, Institut Teknologi Bandung, Bandung.

[6] Pradhana, Bayu Adithya., 2006,Studi dan Implemantasi Persoalan Lintasan Terpendek Suatu Graf dengan Algoritma Dijsktra dan Algoritma Bellman-Ford, *http://mail.informatika.org/~rinaldi/Matdis/2006-2007/Makalah/Makalah0607-26.pdf*, diakses 24 Februari 2013.

[7] Purwananto, Yudhi; Purwitasari, Diana; Wibowo, Agung., 2005, "Implementasi dan Analisis Algoritma Pencarian Rute Terpendek di Kota Surabaya", *JurnalPenelitian dan Pengembangan Telekomunikasi*, No. 2, Vol.10, Desember 2005.

[8] Rachmah, Nur Fajriah., 2008, Aplikasi Algoritma Disjktra dalam Pencarian Lintasan Terpendek Graf, *http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2007-2008/Makalah/MakalahIF2153-0708-113.pdf*, diakses 24 Februari 2013.

[9] Saputra, Ragil., 2011, Sistem Informasi Geografis Pencarian Rute Optimum Obyek Wisata Kota Yogyakarta Dengan Algoritma Floyd-Warshall, *Jurnal Matematika*, No. 1, Vol.14, Hal 19-24, April 2011.

[10] Siang, Jong Jek., 2006, Matematika Diskrit dan Aplikasinya pada Ilmu Komputer Edisi ke-3, Andi, Yogyakarta.

[11] Suherman, Eman; Prasetijo, Agung Budi; Sudjadi., 2011, Simulasi Algoritma Dijkstra pada Protocol Open Shortest Path First,*http://eprints.undip.ac.id/25828/1/ML2F398302.pdf*,diakses 24 Februari 2013.

[12] Sukarto, Haryono., 2006, Transportasi Perkotaan dan Lingkungan, *Jurnal Teknik Sipil*, No. 2, Vol. 3, Juli 2006.

[13] Sukrisno, Ariani Tyas dan Rahman, Arief., 2010, Perancangan Prototype Dynamic Exit Sign dengan Mengembangkat Metode Floyd-Warshall Algorithm Pada Perencanaan Proses Evakuasi Gedung Bertingkat, *http://digilib.its.ac.id/public/ITS-Undergraduate-12955-Paper.pdf*, diakses 24 Februari 2013.