

Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metode Algoritma Dijkstra

(An Application Design for Determining the Shortest Path of Hospital in Purbalingga Using Dijkstra Algorithm Method)

Abdul Ghofur Wibowo¹⁾, Agung Purwo Wicaksono²⁾

^{1) 2)} *Teknik Informatika – F. Teknik – Universitas Muhammadiyah Purwokerto
Jl. Raya Dukuwaluh Purwokerto 53182*

Abstract - Application of Dijkstra's algorithm is an application made to determine the shortest path of hospital in Purbalingga where the starting point comes from a single point, namely the town square of Purbalingga. The system is made with Microsoft Visual Studio 2005 and Microsoft SQL Server 2005. There are two types of hospitals in Purbalingga, such as public hospital and maternal hospital. On the main page, user can specify the type of hospital first before determining the choice of destination hospital. After that, user can know the shortest way to go. There is also a digital map in order to facilitate user into the hospital and information obtained is more informative. On the admin page is used to add data path information, hospital data and map data. Besides, the data path and hospital data can be changed if an error occurs. To determine the shortest path from each hospital, administrator must enter data into the system calculated to form and then saved to the database, which will be used as a user's information can be obtained after choosing the destination hospital.

Keywords: Shortest path, hospital, Dijkstra algorithm.

I. PENDAHULUAN

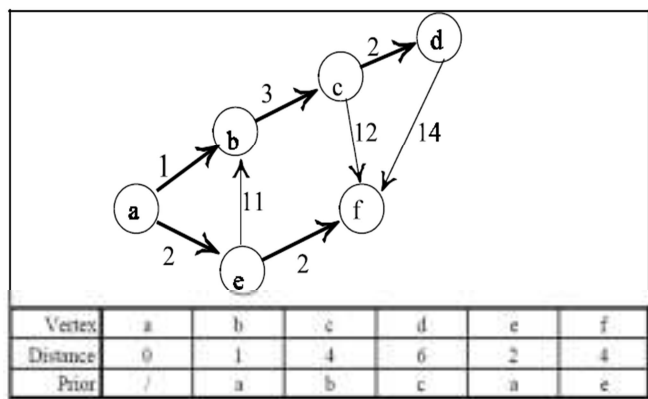
Pada dasarnya manusia membutuhkan waktu untuk mencapai suatu tujuan. Semakin cepat waktu yang ditempuh maka semakin pendek pula jalur yang ditempuh. Hal ini menunjukkan bahwa tingkat efisiensi waktu digunakan pada pola hidup manusia. Apalagi dengan tingkat perkembangan jaman yang begitu cepat, membuat manusia mengalami berbagai macam hambatan dalam menuju suatu lokasi tujuan. Perubahan teknologi membuat manusia dapat mengembangkan pola pikir mereka menjadi lebih baik dari sebelumnya. Suatu lokasi yang dulunya ditempuh dengan waktu yang lama

kini mereka sudah bisa ditempuh dengan waktu yang singkat. Ini disebabkan oleh pola pikir mereka yang semakin berkembang pula. Pemanfaatan berbagai jalur untuk mencapai tujuan merupakan salah satu perkembangan pola pikir manusia.

Jalur menuju rumah sakit dengan jalan tercepat merupakan suatu bentuk kebutuhan manusia. Ini dikarenakan kondisi pasien untuk mendapat pertolongan juga harus cepat dan tepat, sehingga tidak menjadikan hal-hal yang tidak diinginkan terjadi. Letak dari rumah sakit di tiap kota berbeda-beda sehingga jika ingin menuju ke rumah sakit ada banyak pilihan yang harus dipilih, sesuai dengan kebutuhan dari pasien. Jarak dari pusat kota menuju rumah sakit memiliki perbedaan waktu tempuh. Terkadang ada orang yang mengabaikan waktu tempuh untuk menuju rumah sakit dengan jarak terdekat. Oleh karena itu penerapan algoritma sangat diperlukan dalam menyelesaikan kasus ini.

Algoritma adalah sebuah prosedur komputasi yang mentransformasikan sejumlah *input* menjadi sejumlah *output*. Sebuah algoritma dikatakan “benar (*correct*)” jika untuk setiap *input*nya menghasilkan *output* yang benar pula [12]. Dalam hal ini algoritma dapat digunakan sebagai metode untuk mengetahui langkah-langkah secara urut untuk mencapai tujuan. Setiap algoritma memiliki perbedaan dalam mencapai tujuan. Ada beberapa algoritma yang dapat digunakan untuk menentukan jalur terpendek. Salah satunya menggunakan algoritma Dijkstra. Algoritma ini pertama kali dikemukakan oleh Edsger W. Dijkstra pada tahun 1959 dan telah secara luas digunakan dalam menentukan rute tersingkat atau jalur terpendek berdasarkan kriteria tertentu yang digunakan sebagai batasan [13].

Jalur terpendek (*shortest path*) dipublikasikan pada tahun 1959 yang berada di *Numerische Mathematik* yang di edit oleh F.L. Bauer. Pada saat itu algoritma untuk jalur terpendek hampir tidak dianggap. Ada banyak cara untuk pergi dari suatu titik A ke titik B, akan tetapi cara tersebut masih tidak dianggap [9]. Jalur terpendek dapat didefinisikan sebagai masalah kombinatorial dalam grafik dengan bobot terbatas, atau sebagai masalah optimasi yang terus berkelanjutan dalam Geometri Euclidian [7]. Kunci untuk menemukan jalur terpendek adalah dengan mengetahui bagaimana cara menggambarannya, salah satunya menggunakan struktur pohon (Gambar 1).



Gambar 1. Grafik dengan Pohon Jalur Terpendek dalam Sebuah Tabel.

Jalur terpendek dari *node* a ke semua *node* lain digambarkan dengan menunjukkan *node* terakhir yang dikunjungi sebelumnya pada setiap *node*. Sebuah contoh ilustratif digambarkan pada Gambar 1. Dalam gambar tersebut, jalur terpendek dari *node* a ditandai dengan tepi yang lebih gelap [2]. Pada gambar di atas dicontohkan suatu *graph* yang memiliki jarak yang berbeda-beda dari satu titik ke titik lainnya. Diperlukan sebuah algoritma untuk menyelesaikan permasalahan rute pada gambar tersebut, salah satunya menggunakan Algoritma Dijkstra. Algoritma Dijkstra menurut penemunya, seorang ilmuwan komputer, Edsger Dijkstra, adalah sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif. Misalnya, bila *vertices* dari sebuah graf melambangkan kota-kota dan bobot sisi (*edge weights*) melambangkan jarak antara kota-kota tersebut, maka algoritma Dijkstra dapat digunakan untuk menemukan jarak terpendek antara dua kota. *Input* algoritma ini adalah sebuah graf berarah yang berbobot (*weighted directed graph*) *G* dan sebuah sumber *vertices*

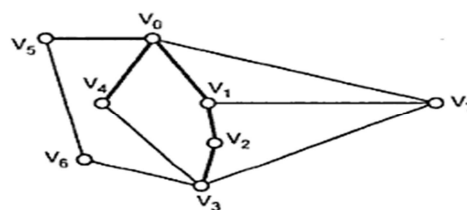
dalam *G* dan *V* adalah himpunan semua *vertices* dalam *graph G*. Setiap sisi dari graf ini adalah pasangan *vertices* (*u,v*) yang melambangkan hubungan dari *vertex* *u* ke *vertex* *v*. Himpunan semua tepi disebut *E*. Bobot (*weights*) dari semua sisi dihitung dengan fungsi:

$$w: E \rightarrow [0, \infty)$$

jadi $w(u,v)$ adalah jarak tak-negatif dari *vertex* *u* ke *vertex* *v*.

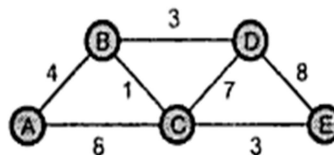
Biaya (*cost*) dari sebuah sisi dapat dianggap sebagai jarak antara dua *vertex*, yaitu jumlah jarak semua sisi dalam jalur tersebut. Untuk sepasang *vertex* *s* dan *t* dalam *V*, algoritma ini menghitung jarak terpendek dari *s* ke *t* [3].

Algoritma Dijkstra adalah algoritma yang populer untuk menentukan jalur terpendek. Algoritma ini disebut sebagai sumber tunggal algoritma untuk menentukan jalur terpendek. Dalam algoritma ini, untuk setiap *vertex* tertentu disebut sebagai titik jalur terpendek ke semua titik yang lain. Dalam algoritma ini tidak hanya terfokus untuk mencari jalur terpendek dari setiap *vertex* tetapi ke semua *vertex* yang tersedia. Algoritma ini dapat diterapkan pada grafik hanya dengan bobot *non-negative*. Algoritma Dijkstra menemukan jalur terpendek ke *graph* ber*vertex* diurutkan jarak dari sumber tertentu. Dalam proses pencarian jalur terpendek, Algoritma Dijkstra menemukan jalur terpendek dari sumber titik *vertex* terdekat menuju titik selanjutnya [11]. Contoh *graph* dapat dilihat pada Gambar 2 berikut.



Gambar 2. Contoh Graph

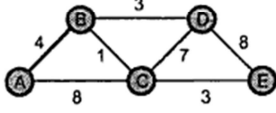
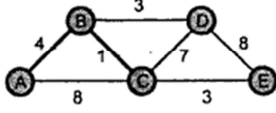
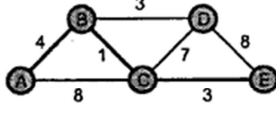
Jalur terpendek dari V_0 telah diperoleh. Pertama temukan jalur terpendek dari $V_0 - V_1$ kemudian $V_1 - V_2$ kemudian dari $V_2 - V_3$ maka jalur terpendek diperoleh (Gambar 3).



Gambar 3. Contoh Jalur yang Memiliki Jarak

Setiap *vertex* sebagai sumber akan menemukan jalur terpendek dari titik satu ke titik yang lainnya dan tampak pada Tabel 1 berikut.

Tabel 1. Penyelesaian pada Sebuah Kasus

Source vertex	Distance with other vertices	Path shown in graph
A	A-B, path = 4 A-C, path = 8 A-D, path = ∞ A-E, path = ∞	
B	B-C, path = 4 + 1 B-D, path = 4 + 3 B-E, path = ∞	
C	C-D, path = 5 + 7 = 12 C-E, path = 5 + 3 = 8	
D	D-E, path = 7 + 8 = 15	

Dari Tabel 1 diperoleh jalur terpendek dari titik A sampai dengan E dan titik itu adalah A – B – C – E dengan panjang jarak = 4 + 1 + 3 = 8. Demikian pula jalur terpendek lainnya dapat diperoleh dengan memilih titik awal dan tujuannya.

Berdasarkan uraian tersebut di atas maka penelitian ini memiliki tujuan membangun aplikasi untuk menentukan jalur terpendek Rumah Sakit di Purbalingga menggunakan metode Algoritma Dijkstra. Manfaat yang diharapkan dari penelitian ini adalah memberikan kemudahan bagi masyarakat untuk menentukan jalur terpendek yang ingin dilalui menuju Rumah Sakit sehingga masyarakat Purbalingga dapat mempersingkat waktu untuk mencapai Rumah Sakit yang dituju.

Pengembangan sistem ini menggunakan *tool* yaitu bahasa pemrograman C# dan *Arcview*. C# adalah bahasa berorientasi objek yang memiliki kode visual menyerupai bahasa pemrograman C++ dan Java. C# memiliki *syntax* yang berbeda dari C++ dalam beberapa hal, tetapi sebagian besar perbedaannya berada pada semantik dan perilakunya, perbedaan tersebut juga berasal dari proses runtime program pada saat dieksekusi [10]. Bahasa pemrograman C# adalah salah satu bahasa pemrograman tingkat menengah dalam membuat program. C# mengisi kesenjangan yang kuat antara C# dan C++. Sebuah file C# menggunakan file berekstensi .cs [4]. C# merupakan bahasa pemrograman yang sederhana, aman, dan mudah. C# dibangun di atas

warisan bahasa C++ dan *programmer* harus segera beradaptasi dengan bahasa C++. C# juga mengkapitalisasi pada kekuatan *.Net Common Language Runtime*, yang menyediakan dukungan *library* yang luas untuk pembuatan pemrograman umum dan aplikasi yang lebih spesifik [6]. Keuntungan dari C# sendiri yaitu *syntax* yang digunakan seperti bahasa C, yang berarti dapat menambah ilmu bahasa pemrograman selain Java, JavaScript atau C++. Saat ini sedikit *programmer* yang mengetahui satu saja bahasa pemrograman, maka dengan menggunakan bahasa pemrograman C# dapat memberikan keuntungan tambahan ilmu bahasa pemrograman lainnya [15].

Pada saat ini segalanya menggunakan *digital*, peta yang digunakan untuk menggambarkan dunia juga sudah berada di dalam komputer. Sekarang dengan ujung jari pun pengguna dapat mencari peta, menemukan objek dan rute, dan merencanakan kegiatan yang terkait dengan peta. Sistem komputer yang memungkinkan untuk menyimpan dan mengakses semua informasi ini dinamakan *Geographic Informations System*. GIS juga merupakan sistem operasional yang memungkinkan mengelola sumber daya untuk menggunakan beberapa alat-alat geografi yang ada. Menggunakan perangkat lunak GIS juga dapat digunakan untuk menempatkan peta dan data geografis lainnya ke dalam komputer. Setelah memiliki data dalam komputer, perangkat lunak GIS juga dapat menyimpan, menampilkan dan mengedit

data yang telah dimasukkan oleh pengguna. Dengan hasil itu pengguna dapat menganalisis dan menghasilkan *output* dalam menggunakan perangkat tersebut [5].

Arcview merupakan sebuah *tools* yang digunakan untuk menyimpan data, grafik dan tabel dalam sebuah *project GIS*. *Project* tersebut digunakan untuk mengatur semua informasi yang dibutuhkan dalam melakukan suatu pembuatan peta. Dalam pembuatan *project* melibatkan lima jenis komponen (*documents*) untuk mengatur informasi: *views*, *tables*, *charts*, *layout* dan *scripts*. Setiap komponen menampilkan data yang berbeda. Setiap komponen memiliki masing-masing fungsi dan terhubung satu dengan yang lain seperti *menus*, *buttons*, and *tools organized* yang memiliki tampilan yang tersendiri [1].

Berikut beberapa hasil penelitian tentang jalur terpendek menggunakan Algoritma Dijkstra:

Pengembangan aplikasi *Traffic Information Systems* yang digunakan untuk menentukan jalur terpendek kereta api di Jerman [14]. Pengguna melakukan permintaan terhadap sistem secara *online* berupa tujuan yang akan ditempuh. Sistem memberikan informasi jalur terpendek secara cepat dan tepat dengan menggunakan Algoritma Dijkstra yang diterapkan pada sistem tersebut, dan disimpulkan bahwa algoritma ini dapat menghasilkan informasi yang optimal.

Pengembangan sebuah aplikasi yang dibuat dalam bentuk *desktop* untuk menentukan sebuah lintasan terpendek [8]. Aplikasi ini dapat memberikan perbandingan antara jalur terpendek menggunakan metode *Greedy* dan Dijkstra. Kegiatan *input* data dilakukan dengan mengisi tujuan dan metode yang akan digunakan. Data diproses dengan menggunakan aplikasi dalam bentuk matrik yang akan menghasilkan sebuah *output* berupa penggambaran jalur yang akan dilalui sesuai dengan algoritmanya. Berdasarkan hasil *output* tersebut, ternyata Algoritma Dijkstra menghasilkan *output* jalur lintasan lebih pendek dari pada Algoritma *Greedy*.

II. METODE

Alat yang digunakan adalah sebuah komputer (PC) bersistem operasi Windows Seven ExtremeEdition RI PC dengan perangkat lunak C# sebagai bahasa pemrogramannya dan MicrosoftSQL Server 2005 sebagai databasenya. Sedangkan hardwarenya berupa speedmeter digital untuk menentukan jarak jalur pada setiap jalan yang dilalui.

A. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data primer berupa hasil survey lapangan dengan

melakukan penelitian jarak pada objek yang bersangkutan. Jarak yang diteliti yaitu dari Alun-alun Purbalingga ke berbagai Rumah Sakit di Purbalingga.

B. Langkah Operasional

Langkah-langkah operasional pembuatan aplikasi ini adalah :

1. Mengumpulkan data jarak jalanyang menuju Rumah Sakit di Purbalingga
2. Merancang database dan query yang diperlukan
3. Merancang aplikasi
4. Membuat aplikasi.
5. Validasi dan verifikasi
6. Implementasi hasil aplikasi.

III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Dalam pembuatan aplikasi ini memiliki kriteria kebutuhan dalam pembuatannya, yaitu:

Satu unit komputer yang digunakan sebagai alat untuk menjalankan aplikasi dengan spesifikasi Processor Pentium Core 2 Duo 2,1 Ghz, RAM 3 Ghz, Hardisk 320 Gb, dan Monitor LCD 14”.

Data nama jalan yang dilalui untuk menuju berbagai rumah sakit di Purbalingga beserta jarak dari masing-masing nama jalan. Satuan yang digunakan dalam jarak suatu jalan menggunakan satuan meter.

1. *Rumah Sakit Ummu Hani* (Tabel 2).

Tabel 2. Jarak Jalan Rumah Sakit Ummu Hani

No	Nama Jalan	Jarak	Simbol
1	Jalan Jambu Karan	250	L
2	Jalan Let. Achmad Nur	350	M
3	Jalan Wirasaba	100	N
4	Jalan Mayjend Panjaitan	100/200	O

2. *Rumah Sakit Panti Nugroho* (Tabel 3).

Tabel 3. Jarak Jalan Rumah Sakit Panti Nugroho

No	Nama Jalan	Jarak	Simbol
1	Jalan Kapten Piere Tendean	200	B
2	Jalan Jend Soedirna	220	H
3	Jalan Puring	400	I
4	Jalan Letnan Suparto	300	J
5	Jalan Letkol Isdiman	10/10/200	K

3. *Rumah Sakit Kasih Ibu* (Tabel 4).

Tabel 4. Jarak Jalan Rumah Sakit Kasih Ibu

No	Nama Jalan	Jarak	Simbol
1	Jalan Onje	300	S
2	Jalan Mayjend Panjaitan	300	O
3	Jalan Pujowiyoto	400	W
4	Jalan Jend Soedirman	1100	H
5	Jalan Raya Penaruban	1500	X
6	Jalan Raya Selakambang	300/2600	Y

4. *Rumah Sakit Harapan Ibu* (Tabel 5).

Tabel 5. Jarak Jalan Rumah Sakit Harapan Ibu

No	Nama Jalan	Jarak	Simbol
1	Jalan Kapten Piere Tendean	200	B
2	Jalan Wiramenggala	500	C
3	Jalan Kom. Notosumarsono	900	D
4	Jalan Achmad Yani	1200	E
5	Jalan G Korakan	1700	F
6	Jalan Mayjend Sungkono	100	G

5. *Rumah Sakit Nirmala* (Tabel 6).

Tabel 6. Jarak Jalan Rumah Sakit Nirmala

No	Nama Jalan	Jarak	Simbol
1	Jalan Kapten Piere Tendean	200	B
2	Jalan Wiramenggala	500	C
3	Jalan Kom. Notosumarsono	900	D
4	Jalan Achmad Yani	1200	E
5	Jalan G Korakan	1700	F
6	Jalan Gunung Kraton	700	P
7	Jalan Letnan Yusuf	900/1600	Q

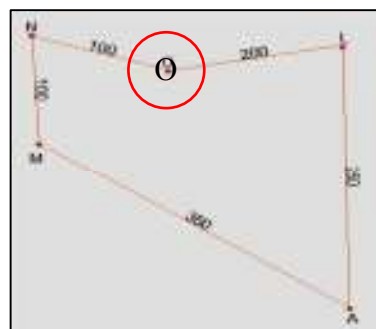
6. *Rumah Sakit Wirasana* (Tabel 7).

Tabel 7. Jarak Jalan Rumah Sakit Wirasana

No	Nama Jalan	Jarak	Simbol
1	Jalan Let. Achmad Nur	350	M
2	Jalan Wirasaba	100	N
3	Jalan Mayjend Panjaitan	100	O
4	Jalan A W Soemarmo	900	R
5	Jalan Onje	600	S
6	Jalan Veteran	300	T
7	Jalan Ketuhu	900	U
8	Jalan Tentara Pelajar	200/1700	V

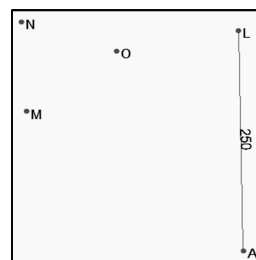
B. *Langkah pencarian jalur terpendek pada setiap rumah sakit*

Pada setiap rumah sakit memiliki jalur terpendek yang dapat dilalui dengan menggunakan Algoritma Dijkstra. Berikut ini adalah penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan, dengan contoh obyek Rumah Sakit Ummu Hani. Node awal A, Node tujuan O. Setiap edge yang terhubung antar node telah diberi nilai (Gambar 4).



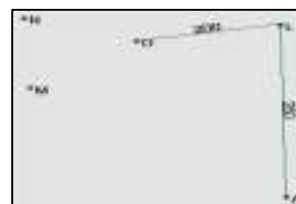
Gambar 4. Contoh Kasus Dijkstra Rumah Sakit Ummu Hani - Langkah 1

Dijkstra melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node*A), dan hasil yang didapat adalah *node* L karena bobot nilai *node* L paling kecil dibandingkan nilai pada *node* lain, nilai = 250 (Gambar 5).



Gambar 5. Contoh Kasus Dijkstra Rumah Sakit Ummu Hani - Langkah 2

Node L diset menjadi *node* keberangkatan dan ditandai sebagai *node* yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap *node-node* tetangga yang terhubung langsung dengan *node* yang telah terjamah. Dan kalkulasi dijkstra menunjukkan bahwa *node*O yang menjadi *node* keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai $250 + 200 = 450$ (Gambar 6).

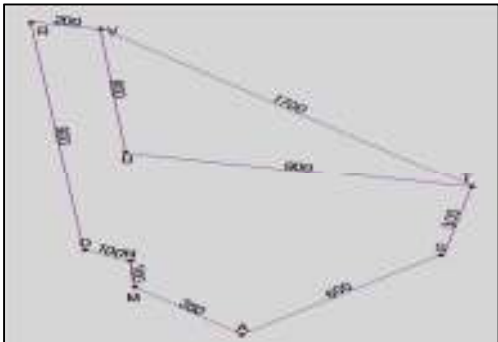


Gambar 6. Contoh Kasus Dijkstra Rumah Sakit Ummu Hani - Langkah 3

Node O telah tercapai lewat *node*L. Jalur yang dapat dilalui adalah A – L – O dengan jarak $250 + 200$ dan nilai bobot yang didapat adalah 450. Bila *node* tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.

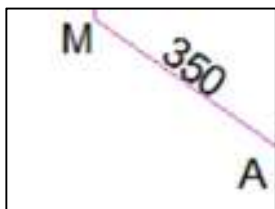
Dijkstra kembali melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node* A), dan melakukan pengecekan kepada tiap *node*. Setelah pengecekan selesai, ternyata hanya jalur A – L – O yang memiliki jalur terpendek dari semua *node* yang terhubung, dan proses kalkulasi dijkstra dinyatakan selesai.

Contoh lain adalah kasus Rumah Sakit Wirasana. *Node* awal A, *Node* tujuan V. Setiap *edge* yang terhubung antar *node* telah diberi nilai (Gambar 7).



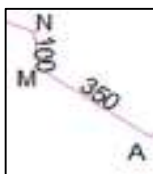
Gambar 7. Contoh Kasus Dijkstra Rumah Sakit Wirasana- Langkah 1

Dijkstra melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node*A), dan hasil yang didapat adalah *node*M karena bobot nilai *node*M paling kecil dibandingkan nilai pada *node* lain, nilai = 350 (0+350) (Gambar 8).



Gambar 8. Contoh Kasus Dijkstra Rumah Sakit Wirasana- Langkah 2

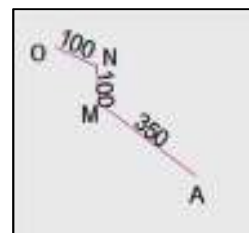
Dijkstra melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node*M), dan hasil yang didapat adalah *node*N karena bobot nilai *node*N paling kecil dibandingkan nilai pada *node* lain, nilai = 450 (350+100) (Gambar 9).



Gambar 9. Contoh Kasus Dijkstra Rumah Sakit Wirasana- Langkah 3

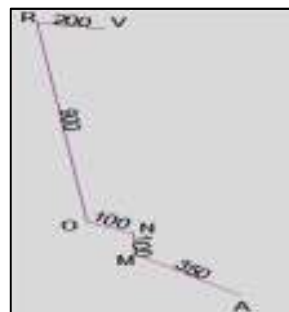
Dijkstra melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan

(*node*M), dan hasil yang didapat adalah *node*O karena bobot nilai *node*O paling kecil dibandingkan nilai pada *node* lain, nilai = 550 (450+100) (Gambar 10).



Gambar 10. Contoh Kasus Dijkstra Rumah Sakit Wirasana- Langkah 4

Dijkstra melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node*M), dan hasil yang didapat adalah *node*O karena bobot nilai *node*O paling kecil dibandingkan nilai pada *node* lain, nilai = 1450 (550+900) (Gambar 11).



Gambar 11. Contoh Kasus Dijkstra Rumah Sakit Wirasana- Langkah 5

- a. *Node*V telah tercapai lewat *node*R. Jalur yang dapat dilalui adalah A – M – N– O – R - V dengan jarak 1450+ 200 dan nilai bobot yang didapat adalah 1650. Bila *node* tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai (Gambar 12).



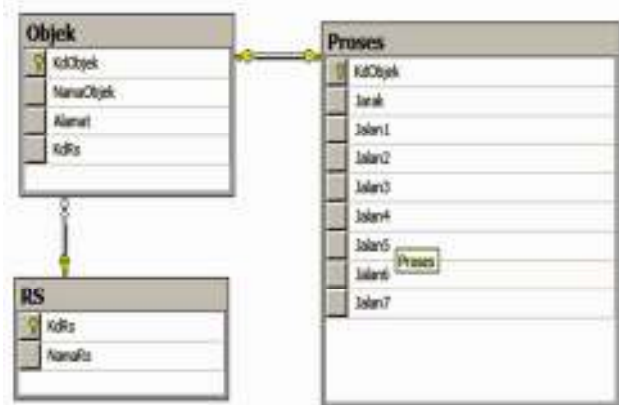
Gambar 12. Contoh Kasus Dijkstra Rumah Sakit Wirasana- Langkah 6

Dijkstra kembali melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node*A), dan melakukan pengecekan

kepada tiap *node*. Setelah pengecekan selesai, ternyata hanya jalur A – M – N– O – R - V yang memiliki jalur terpendek dari semua *node* yang terhubung, dan proses kalkulasi dijkstra dinyatakan selesai.

C. Perancangan Database

Sebelum membuat aplikasi terlebih dahulu membuat *database*, berikut rancangan *database* dari aplikasi ini (ambar 13).



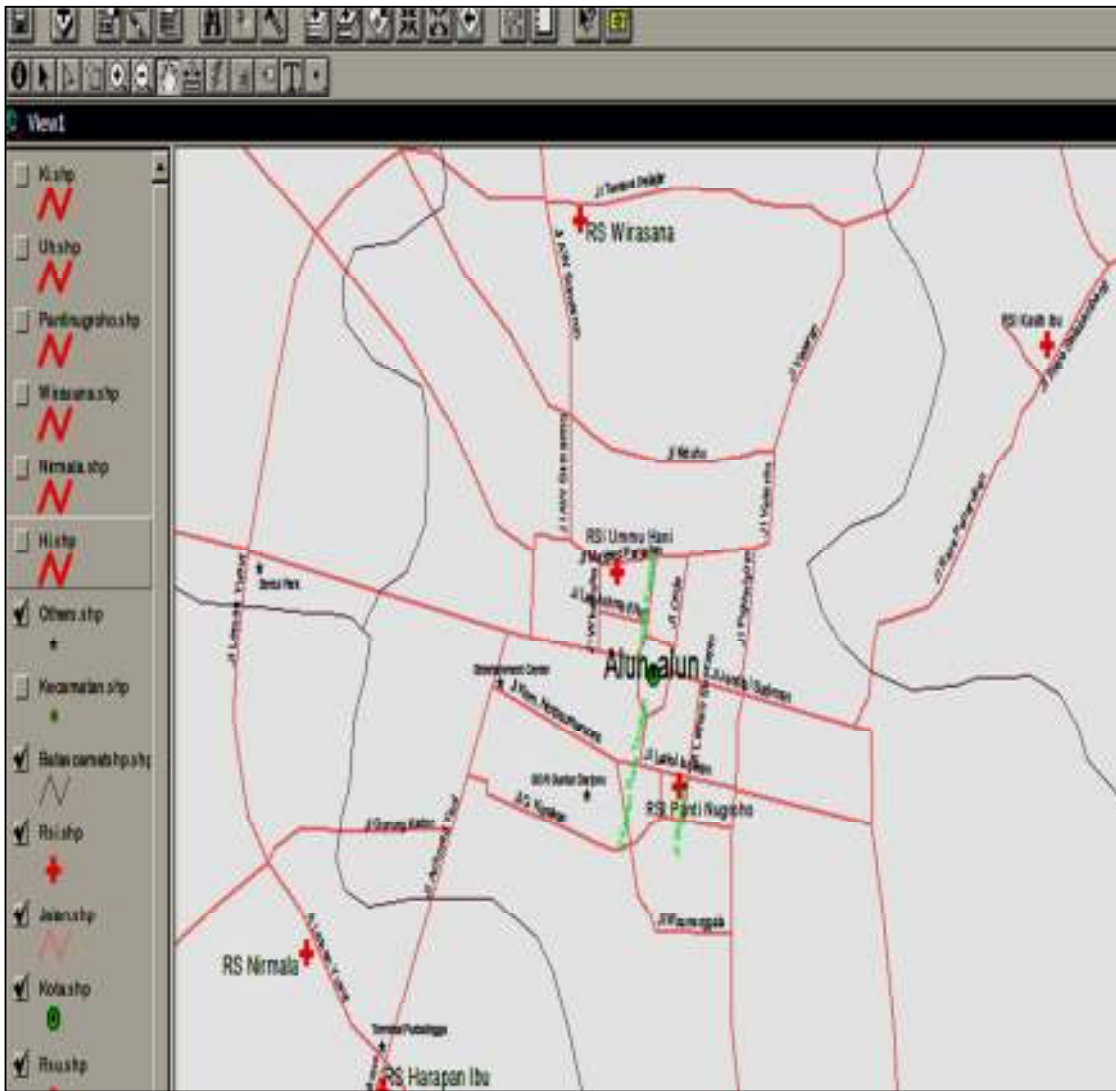
Gambar 13. DatabaseDiagram

D. Perancangan Peta

Pada pembuatan aplikasi ini diperlukan peta jalan yang digunakan agar data yang diterima pengguna

setelah menjalankan aplikasi ini lebih informatif. Sebelum membuat peta diperlukan data peta seperti data jalan dan data rumah sakit. Data peta tersebut digunakan dalam proses digitasi dan atribut. Sebelum membuat peta pada ArcView terlebih dahulu mengatur Map dan Distance unit pada peta yang akan dibuat. Proses digitasi menggunakan peta digital daerah Purbalingga yang kemudian dirubah dalam bentuk shppada Arcview. Kemudian proses digitasi dilakukan sesuai dengan peta digital daerah Purbalingga, setelah itu dilanjutkan dengan proses editing peta.

Pada saat proses digitasi selesai kemudian dilakukan proses pemberian atribut pada Rumah Sakit, Jalan, Alun-alun, Gelora Goentur Darjono, Sentul Park, Entertainment Center, dan Terminal Purbalingga. Proses pemberian atribut ini disesuaikan dengan hasil digitasi yang dilakukan. Penambahan field pada proses atribut digunakan agar data proses digitasi memiliki keterangan nama. Proses digitasi dan atribut yang telah selesai, dapat dilihat pada Gambar 14 berikut ini.



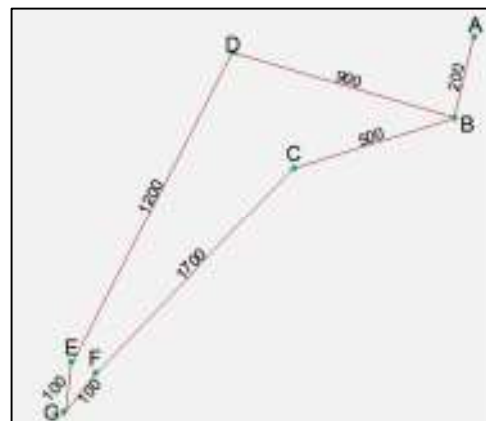
Gambar 14. Proses Digitasi dan Atribut

E. Perancangan matrik dan jalur pada masing-masing rumah sakit.

Perancangan matrik dan jalur ini diimplementasikan dengan menggunakan tabel dan gambar jalur menuju rumah sakit. ada tabel matrik terdapat huruf yang digunakan sebagai inisial jalan yang sesungguhnya. Berikut ini beberapa beberapa jalur yang digambarkan menggunakan inisial huruf dari state awal alun-alun Purbalingga dan tabel matrik dari gambar tersebut. Sebagai contoh kasus Rumah Sakit Harapan Ibu dan Rumah Sakit Wirasana.

1. Rumah Sakit Harapan Ibu

Rancangan jalur menuju Rumah Sakit Harapan Ibu dapat dilihat pada Gambar 15. Sedangkan tabel matrik dari jalur tersebut tampak pada Tabel 8 berikut ini:



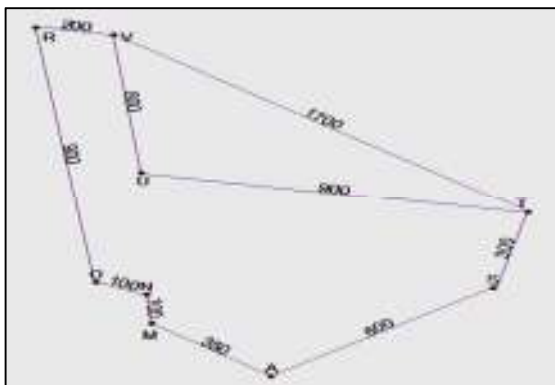
Gambar 15. Jalur Rumah Sakit Harapan Ibu

Tabel 8. Tabel Matrik Jalur Rumah Sakit Harapan Ibu

	A	B	C	D	E	F	G
A	0	100	x	x	x	x	x
B	100	0	500	900	x	x	x
C	x	500	0	x	x	1700	x
D	x	900	x	0	1200	x	x
E	x	x	x	1200	0	x	100
F	x	x	x	x	x	0	100
G	x	x	x	x	100	100	0

2. Rumah Sakit Wirasana

Rancangan jalur menuju Rumah Sakit Wirasana dapat dilihat pada Gambar 16. Sedangkan tabel matrik dari jalur tersebut tampak pada Tabel 9 berikut ini:



Gambar 16. Jalur Rumah Sakit Wirasana

Tabel 9. Tabel Matrik Jalur Rumah Sakit Wirasana

	A	M	N	O	R	S	T	U	V
A	0	350	x	x	x	600	x	x	x
M	350	0	100	x	x	x	x	x	x
N	x	100	0	100	x	x	x	x	x
O	x	x	100	0	900	x	x	x	x
R	x	x	x	900	0	x	x	x	200
S	600	x	x	x	x	0	300	x	x
T	x	x	x	x	x	300	0	900	1700
U	x	x	x	x	x	x	x	0	800
V	x	x	x	x	200	x	1700	800	0

F. Pengujian Aplikasi

Aplikasi dihosting pada layanan hosting gratis dengan nama layanan provider **aspspider.com**. Kelemahan dari hosting gratis ini yaitu masa hosting hanya dibatasi sampai dengan 90 hari dari sejak membuat nama websitenya. Nama website yang dapat diakses untuk menjalankan aplikasi ini yaitu **aspspider.info/Gofuy**. Pengujian aplikasi ini diimplementasikan dalam bentuk tabel dan interface seperti pada Tabel 10 dan Gambar 17 sampai dengan Gambar 25.

Tabel 10. Tabel Pengujian

Tipe Nilai	Input		Output			KET
	Jenis Rumah Sakit	Nama Rumah Sakit	Alamat	Jarak	Jalan yang dilalui	
Valid	Rumah Sakit Umum	Rumah Sakit Harapan Ibu	Jalan Maviend Sungkono No 1	2400 m	Jl. Kapten Piere Tendean - Jl. Kom. Notosumarsono - Jl. Ahmad Yani - Jl. Maviend Sungkono	Berhasil
Valid	Rumah Sakit Umum	Rumah Sakit Nirmala	Jalan Letnan Yusuf No 17	3200 m	Jl. Kapten Piere Tendean - Jl. Kom. Notosumarsono - Jl. Ahmad Yani - Jl. Letnan Yusuf	Berhasil
Valid	Rumah Sakit Umum	Rumah Sakit Wirasana	Jalan Tentara Pelajar No 22	1650 m	Jl. Achmad Nur - Jl. Wirasaba - Jl. Maviend Panaitan - Jl. AW Sumarmo - Jl. Tentara Pelajar	Berhasil
Valid	Rumah Sakit Ibu	Rumah Sakit Panti Nugroho	Jalan Letkol Isdiman No 20	400 m	Jl. Kapten Piere Tendea - Jl. Letkol Isdiman	Berhasil
Valid	Rumah Sakit Ibu	Rumah Sakit Ummu Hani	Jalan Maviend Panaitan No 40 A	450 m	Jl. Jambu Karang - Jl. Maviend Panaitan	Berhasil
Valid	Rumah Sakit Ibu	Rumah Sakit Kasih Ibu	Jalan Raya Selakambang No 27	2900 m	Jl. Jendral Soedirman - Jl. Raya Penaruban - Jl. Raya Selakambang	Berhasil

Halaman utama (Gambar 17) merupakan halaman utama saat aplikasi dijalankan dan sekaligus sebagai halaman yang digunakan oleh pengguna dalam menentukan jalur terpendek.



Gambar 17. Form Utama

Halaman admin (Gambar 18) merupakan halaman yang digunakan oleh admin saat masuk ke sistem dan sekaligus sebagai halaman untuk menambah data jalan, data rumah sakit dan data peta. Berikut adalah tampilan halaman admin.



Gambar 18. Form Admin

Halaman Algoritma Dijkstra dari salah satu rumah sakit di Purbalingga yaitu Rumah Sakit Ummu Hani. Di sini sistem perhitungan Algoritma Dijkstra dilakukan

berdasarkan tabel matrik yang telah dibuat pada script dijkstra yang bersangkutan. Berikut tampilan dari halaman Ummu Hani yang tampak pada Gambar 19.

Gambar 19. Form Ummu Hani

1. Rumah Sakit Umum

- a. Rumah Sakit Harapan Ibu. Pengujian aplikasi dengan objek Rumah Sakit Harapan Ibu tampak pada Gambar 20 berikut ini.



Gambar 20. Form Hasil Rumah Sakit Harapan Ibu

- b. *Rumah Sakit Nirmala*. Pengujian aplikasi dengan objek Rumah Sakit Nirmala tampak pada Gambar 21 berikut ini.



Gambar 21. *Form Hasil Rumah Sakit Nirmala*

- c. *Rumah Sakit Wirasana*. Pengujian aplikasi dengan objek Rumah Sakit Wirasana tampak pada Gambar 22 berikut ini.



Gambar 22. *Form Hasil Rumah Sakit Wirasana*

2. *Rumah Sakit Ibu*

- a. *Rumah Sakit Panti Nugroho*. Pengujian aplikasi dengan objek Rumah Sakit Panti Nugroho tampak pada Gambar 23 berikut ini.



Gambar 23. Form Hasil Rumah Sakit Panti Nugroho

- b. *Rumah Sakit Ummu Hani*. Pengujian aplikasi dengan objek Rumah Sakit Ummu Hani tampak pada Gambar 24 berikut ini.



Gambar 24. Form Hasil Rumah Sakit Ummu Hani

- c. *Rumah Sakit Kasih Ibu*. Pengujian aplikasi dengan objek Rumah Sakit Kasih Ibu tampak pada Gambar 25 berikut ini.



Gambar 25. Form Hasil Rumah Sakit Kasih Ibu

IV. PENUTUP

A. Simpulan

Simpulan dari penelitian ini adalah telah dihasilkan sebuah aplikasi penentu jalur terpendek menggunakan Algoritma Dijkstra, yang dapat digunakan di kota Purbalingga dengan titik awal berasal dari Alun-alun Purbalingga untuk menuju rumah sakit yang ada di daerah tersebut. Aplikasi ini diharapkan akan dapat memudahkan pengguna yang akan menuju rumah sakit dengan cepat dan mengerti jalur mana saja yang harus dilalui. Dengan adanya peta yang dapat menjelaskan jalur mana saja yang harus dilalui membuat aplikasi ini lebih informatif dan tepat.

B. Saran

Terdapat saran yang dapat disampaikan dalam proses pengembangan aplikasi ini yaitu jarak jalan yang dilalui masih diolah dalam script dijkstra yang terdapat dalam sistem. Ini dikarenakan untuk mengurangi kesalahan jarak jalur terpendek. Pada aplikasi ini jika terdapat persimpangan data harus diolah terlebih dahulu sebelum dimasukkan kedalam script dijkstra.

DAFTAR PUSTAKA

[1] Breslin, P., Frunzi, N., Napoleon, E., Ormsby, T., 1996, *Getting to know ArcView GIS*, ESRI, California.

[2] Cargal, J. M., 2005, *Discrete Mathematics for Neophytes*, tersedia pada <http://www.cargalmathbooks.com/lectures.htm>, diakses pada tanggal 10 Januari 2012.

[3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., 2001, *Introduction to Algorithms, Second Edition*, MIT Press, Cambridge.

[4] Davis, S. R dan Sphar, C., 2006, *C# 2005 for Dummies*, Wiley Publishing, Inc., Indianapolis.

[5] Demers, M. N., 2009, *GIS for Dummies*, Wiley Publishing, Inc., Indianapolis.

[6] Gunnerson, E., 2001, *A Programmer's Introduction to C#*, Appres, New York.

[7] Klette, R dan Lie F., 2011, *Euclidean Shortest Paths*, Springer, London.

[8] Lubis, H. S., 2009. *Perbandingan Algoritma Greedy dan Dijkstra untuk menentukan lintasan terpendek*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, Medan.

[9] Misa, T. J., 2010, An Interview with Edsger W. Dijkstra, *Viewpoints*, 41-47.

[10] Nash, T, 2010, *Accelerated C# 2010*, Appres, New York.

[11] Puntambekar, A.A., 2008, *Analysis and Design of Algorithms*, Technical Publications Pune, India.

[12] Purbasari, I. Y., 2007, *Desain dan Analisis Algoritma*, Graha Ilmu, Yogyakarta.

- [13] Purboyo, A. H., 2010, Aplikasi Algoritma Dijkstra – Rute Tersingkat (*Shortest Path*), tersedia pada <http://civilinc.blogspot.com/2010/01/aplikasi-algoritma-dijkstra-rute.html>, diakses pada tanggal 7 Desember 2011.
- [14] Schulz, F., Wagner, D., Weihe, K., 2000, Dijkstra's algorithm on-line: an empirical case study from public railroad transport, *Journal of Experimental Algorithmics (JEA)*, ISSN: 1084 – 6654, Vol 5, No 12.
- [15] Tabor, B., 2005, "Beginning Visual Basic 2005 Express Edition Video Series" & "Beginning Visual C# 2005 Express Edition Video Series", tersedia pada <http://lab.msdn.microsoft.com/express/beginner>, diakses pada tanggal 14 Desember 2011.
- [16] Wasson, C. S., 2006, *System Analysis, Design, and Development Concept, Principles, and Practices*, a John Wiley & Sons, Inc., New Jersey.