

Penerapan Metode Dijkstra Pencarian Rute Terpendek Sekolah Luar Biasa (SLB) di Kota Medan

Anisa Pitri

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338 Medan, Indonesia

Abstrak

Kota Medan merupakan salah satu daerah yang memiliki potensi pendidikan yang beranekaragam dan berkualitas. Dengan potensi yang ada segala upaya telah dilakukan oleh pemerintah provinsi khususnya Dinas kota medan untuk mempromosikan dan memajukan dunia pendidikan khususnya pendidikan di Sekolah Luar Biasa. Oleh karena itu diperlukan sistem informasi geografis yang menghasilkan informasi petunjuk jalan yang paling efektif ditempuh untuk sampai ke tujuan melakukan aktivitas – aktivitas tersebut. Salah satu metode yang dapat digunakan dalam pencarian rute terpendek adalah algoritma Dijkstra. Algoritma Dijkstra menemukan rute terpendek dari sebuah simpul spesifik menuju simpul spesifik lainnya dan jika diteruskan maka akan memberikan rute terpendek dari simpul spesifik menuju seluruh simpul – simpul lain di dalam sebuah graf. Pembuatan SIG untuk mengetahui letak Sekolah Luar Biasa berbasis web dapat membantu atau sebagai teknologi alternatif dalam perkembangan dunia pendidikan itu sendiri. Saat ini web merupakan salah satu sumber informasi yang banyak dipakai untuk sarana promosi bagi sekolah yang ada di suatu daerah. Melalui teknologi gis, maka perlu dikembangkan sebuah aplikasi yang mampu menyediakan informasi jalan untuk Kota medan yang dapat menunjukkan rute terpendek perjalanan menuju sekolah luar biasa di kota medan.

Kata Kunci : Algoritma Dijkstra, Sistem Informasi Geografis, SLB.

Abstract

Medan City is one of the regions that has diverse and quality educational potential. With the potential that exists, all efforts have been made by the provincial government, especially the Medan City Office to promote and advance the world of education, especially education in Special Schools. Therefore a geographic information system is needed which produces the most effective information on road guidance to get to the destination of these activities. One method that can be used in finding the shortest route is Dijkstra's algorithm. Dijkstra's algorithm finds the shortest route from a specific node to another specific node and if it continues it will give the shortest route from the specific node to all other vertices in a graph. Making GIS to find out the location of web-based Extraordinary Schools can help or as an alternative technology in the development of the world of education itself. At present the web is one source of information that is widely used for promotional facilities for schools in an area. Through GIS technology, it is necessary to develop an application that is able to provide road information for Medan City which can show the shortest route to the special school in Medan.

Keywords: Dijkstra Algorithm, Geographic Information System, SLB.

1. PENDAHULUAN

Kebutuhan akan fasilitas pendidikan untuk anak berkebutuhan khusus (ABK) membuat tersebarnya sekolah luar biasa (SLB) di kota Medan. Sekolah Luar Biasa (SLB) merupakan sekolah yang sesuai bagi anak dengan karakteristik khusus yang berbeda dengan anak pada umumnya tanpa selalu menunjukkan pada ketidakmampuan mental, emosi atau fisik. Semakin banyak sekolah luar biasa maka pilihan untuk menyekolahkan anak berkebutuhan khusus semakin banyak. Dengan banyaknya sekolah luar biasa (SLB) di kota medan, maka akan menyulitkan masyarakat umum terutama orang tua anak berkebutuhan khusus dalam memilih rute terpendek yang akan dituju agar dapat menghemat waktu perjalanan. Salah satu cara untuk mengatasi masalah tersebut yakni dengan menggunakan algoritma Dijkstra sebagai salah satu metode pencarian rute terpendek untuk sampai ke tempat tujuan dengan cepat.

Algoritma Dijkstra merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip dari algoritma Dijkstra adalah dengan pencarian dua lintasan yang paling kecil. Algoritma Dijkstra merupakan salah satu varian dari algoritma greedy, yaitu salah satu bentuk algoritma populer dalam persoalan yang terkait dengan masalah optimasi. Sifatnya sederhana dan lempang. Sesuai dengan artinya yang secara harafiah berarti tamak atau rakus namun tidak dalam konteks negatif. Algoritma greedy ini hanya memikirkan solusi terbaik yang akan diambil pada. Prinsipnya, ambillah apa yang bisa anda dapatkan saat ini. Keputusan yang telah diambil pada setiap langkah tidak akan bisa di ubah kembali. Intinya algoritma greedy ini berupaya membuat pilihan nilai optimum lokal pada setiap langkah dan berharap agar nilai optimum lokal ini mengarah kepada nilai optimum global.

2. LANDASAN TEORI

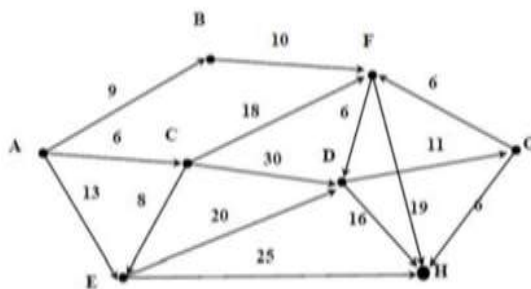
2.1 Algoritma Dijkstra

Algoritma Dijkstra merupakan salah satu varian dari algoritma Greedy, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Sesuai dengan artinya yang secara harafiah berarti tamak atau rakus namun tidak dalam konteks negatif. Dalam mencari solusi, algoritma Dijkstra menggunakan prinsip Greedy, yaitu mencari solusi optimum pada setiap langkah yang dilalui, dengan tujuan untuk mendapatkan solusi optimum pada langkah selanjutnya yang akan mengarah pada solusi terbaik. Cara kerja algoritma Dijkstra juga menggunakan prinsip antrian, akan tetapi antrian yang digunakan algoritma Dijkstra adalah antrian berprioritas. Jadi hanya simpul yang memiliki prioritas tertinggi yang akan ditelusuri. Dalam menentukan simpul yang berprioritas, algoritma ini membandingkan setiap nilai (bobot) dari simpul yang berada pada satu level. Selanjutnya nilai (bobot) dari setiap simpul tersebut disimpan untuk dibandingkan dengan nilai yang akan ditemukan dari rute yang baru ditemukan kemudian, begitu seterusnya sampai ditemukan simpul yang dicari [1].

Algoritma Dijkstra merupakan algoritma yang paling sering digunakan dalam pencarian rute terpendek, sederhana penggunaannya dengan menggunakan simpul-simpul sederhana pada jaringan jalan yang tidak rumit [2]. Adapun nama algoritma Dijkstra sendiri berasal dari penemunya yaitu Edsger Dijkstra.

Dalam mencari solusi, algoritma Dijkstra menggunakan prinsip greedy yaitu mencari solusi optimum pada setiap langkah yang dilalui, dengan tujuan untuk mendapatkan solusi optimum pada langkah selanjutnya yang akan mengarah pada solusi terbaik. Hal ini membuat kompleksitas waktu algoritma Dijkstra menjadi cukup besar.

Cara kerja algoritma Dijkstra hampir sama dengan cara kerja algoritma Breadth-First Search (BFS) yaitu dengan menggunakan prinsip antrian (queue), akan tetapi antrian yang digunakan algoritma Dijkstra adalah antrian berprioritas (priority queue). Jadi hanya simpul yang memiliki prioritas tertinggi yang akan ditelusuri. Dalam menentukan simpul yang berprioritas algoritma ini membandingkan setiap nilai (bobot) dari simpul yang berada pada satu level, selanjutnya nilai (bobot) dari setiap simpul tersebut disimpan untuk dibandingkan dengan nilai yang akan ditemukan dari rute yang baru ditemukan kemudian, begitu seterusnya sampai ditemukan simpul yang dicari. Jika menggunakan algoritma Dijkstra untuk menentukan jalur terpendek dari suatu graph, maka akan menemukan jalur yang terbaik karena pada waktu penentuan jalur yang akan dipilih akan dianalisis bobot dari node yang belum terpilih, lalu dipilih node dengan bobot terkecil. Algoritma Dijkstra mencari jarak terpendek dari node asal ke node terdekatnya, kemudian ke node kedua, dan seterusnya. Contoh penerapan node pencarian jalur terpendek dari dilihat pada gambar berikut ini :



Gambar 1. Penerapan Node Pencarian Jalur Terpendek

Misalnya diberikan graph berbobot dan berarah seperti gambar di atas. Akan dicari lintasan terpendek dari simpul A ke semua simpul lain. Jadi, lintasan terpendek dari:

- A ke C adalah A, C dengan panjang = 6.
- A ke B adalah A, B dengan jarak = 9.
- A ke E adalah A, B, E dengan jarak = 13.
- A ke F adalah A, B, F dengan jarak = 19.
- A ke D adalah A, B, F, D dengan jarak = 25.
- A ke G adalah A, B, F, D, G dengan jarak = 36.
- A ke H adalah A, B, F, H dengan jarak = 38.

2.2 Tinjauan Umum Sekolah Luar Biasa (SLB)

Sekolah Luar biasa (SLB) tidak luput dari anak-anak berkebutuhan khusus (ABK) anak yang Berkebutuhan Khusus (ABK) merupakan istilah lain untuk menggantikan kata Anak Luar Biasa (ALB) yang menandakan adanya kelainan khusus yang memiliki karakteristik berbeda antara satu dengan yang lainnya.

Anak berkebutuhan khusus (ABK) terdiri atas beberapa kategori. Kategori cacat A (tunanetra) ialah anak dengan gangguan penglihatan, kategori cacat B (tunawicara dan tunarungu) ialah anak dengan gangguan bicara dan gangguan pendengaran. Kategori ini dijadikan satu karena biasanya antara gangguan bicara dan gangguan pendengaran terjadi dalam satu keadaan, kategori cacat C (tunagrahita) ialah anak dengan gangguan intelegensi rendah atau perkembangan kecerdasan yang terganggu, kategori cacat D (tunadaksa) ialah anak dengan gangguan

pada tulang dan otot yang mengakibatkan terganggunya fungsi motorik, kategori cacat tunalaras ialah anak dengan gangguan tingkah laku sosial yang menyimpang, kategori anak berbakat ialah anak dengan keunggulan dan kemampuan berlebih (IQ tinggi), dan kategori anak berkesulitan belajar ialah anak dengan ketidakberfungsian otak minimal [3].

3. ANALISA DAN PEMBAHASAN

Sekolah luar biasa merupakan salah satu pendidikan yang sangat dibutuhkan anak berkebutuhan khusus dimana anak berkebutuhan khusus juga berhak untuk mendapatkan pendidikan yang layak untuk mengembangkan potensi mereka walaupun dengan kekhususan yang mereka miliki. Dengan banyaknya Sekolah Luar Biasa (SLB) di Kota Medan, maka akan cukup membingungkan seseorang dalam memilih rute terpendek yang akan dituju agar dapat menghemat waktu perjalanan. Sehingga pencarian rute sekolah luar biasa (SLB) terpendek merupakan salah satu permasalahan yang penting, karena dapat membantu masyarakat dalam menentukan pilihan rute mana yang dapat menghemat waktu perjalanan..

3.1 Penerapan Pencarian Jalur Terpendek Algoritma Dijkstra

Penerapan Metode *Dijkstra* dalam pencarian rute terpendek SLB di Kota Medan adalah pada pencarian jarak lokasi terdekat dari tempat SLB awal ke SLB tujuan. Sehingga didapat rute terpendek menuju lokasi tujuan tersebut. Sebelum menerapkan metode *Dijkstra* di Kota Medan terlebih dahulu di terangkan tentang jumlah SLB dan juga jarak antara satu SLB ke SLB lainnya. Berikut adalah daftar lokasi SLB yang ada di Kota Medan yang menjadi *entry point* dijabarkan dalam tabel berikut :

Tabel 1. Daftar Nama SLB di Kota Medan

No	Nama Sekolah	Alamat
1	SLB - BC PERGURUAN AL- AZHAR MEDAN	JL. Pintu Air IV No : 214 Padang Bulan Medan. Kwala Bekala Kec. Medan Johor
2	SLB ABC TPI MEDAN	JL. Sm. Raja Km. 7 N0 : 5 Medan arjosari -1 Kec. Medan Amplas
3	SLB - A KARYA MURNI MEDAN	Jl. Karya Wisata No : 6 Medan Johor Gedung Johor Kec. Medan Johor
4	SLB - B KARYA MURNI MEDAN	Jl. H.M Joni No : 66 A. Teladan Timur Kec Kota Medan
5	SLB – BC MARKUS MEDAN	Jl. Kapten Muslim No : 226 Helvetia Timur Kec. Medan Helvetia
6	SLB – C ABDI KASIH	Jl. Rawe IV (Psr 6) No : 139 Tangkahan Kel Tangkahan. Kec. Medan Labuhan
7	SLB – C MUZDALIFAH	Jl. Guru IV, Gg Merak No : 15 A, Medan Kec. Kota Medan
8	SLB – C SANTA LUSIA MEDAN	Jl. Sindoru No : 04 Pusat Pasar Medan. Kec Kota Medan
9	SLB – C & D YPAC MEDAN	Jl. Adinegoro No : 2 Medan Gaharu Kec. Medan Timur
10	SLB – E NEGERI PEMBINA MEDAN	Jl. Karya Ujung Medan Helvetia Timur Kec. Helvetia
11	SLB PONDOK KASIH	Jl. Mesjid No.4, Cinta Damai, Medan Helvetia, Medan
12	YAYASAN KARYA MURNI	JL. Bahagia, No. 2, 20217, Teladan Tim., Medan Kota
13	YAYASAN ABDI KASIH	Jl. A. Rivai No.14, Madras Hulu, Medan Polonia,
14	SLB-CATOLIK KARYA TULUS MEDAN	Jl. Mesjid No.4, Cinta Damai, Medan Helvetia, Medan

Pada penerpan Algoritma *Dijkstra* yang telah diimplementasikan ke dalam sebuah aplikasi akan dipergunakan dalam mencari rute terpendek untuk menuju lokasi SLB yang menjadi lokasi tujuan. Namun karena keterbatasan data yang ada, maka *sample* data yang dipergunakan hanya 14 tempat (*verteks/node*) yakni dengan masing-masing memiliki bobot pada *link*-nya (lokasi) seperti tabel berikut :

Tabel 2. Daftar Nama SLB di Kota Medan

LOKASI	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A		6	6	12	∞	∞	∞	∞	∞	∞	∞	∞	6	∞
B	6		∞	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C	6	∞		∞	∞	7	∞	∞	∞	∞	7	∞	6	∞
D	12	6	∞		3	5	∞	∞	∞	∞	∞	∞	∞	∞
E	∞	∞	∞	3		5	18	∞	13		∞	∞	∞	∞
F	∞	∞	7	5	5		∞	∞	∞	∞	∞	∞	∞	∞
G	∞	∞	∞	∞	18	∞		16	∞	∞	∞	15	∞	∞

H	∞	∞	∞	∞	∞	∞	16	∞	6	17	∞	2	∞	∞
I	∞	∞	∞	∞	13	∞	∞	6	∞	∞	∞	∞	∞	∞
J	∞	∞	∞	∞	∞	∞	∞	17	∞	∞	9	5	∞	7
K	∞	∞	7	∞	∞	∞	∞	∞	∞	9	∞	∞	14	∞
L	∞	∞	∞	∞	∞	∞	15	2	∞	∞	∞	∞	∞	6
M	6	∞	6	∞	∞	∞	∞	∞	∞	5	14	∞	∞	∞
N	∞	∞	∞	∞	∞	∞	∞	∞	∞	7	∞	6	∞	∞

Dari Tabel 2 diatas Angka angka yang ada diibaratkan jarak (Kilo Meter) antara dari satu sekolah kesekolah lainnya yang digunakan sebagai acuan dalam penghitungan rute terpendek pada lokasi SLB di Kota Medan menggunakan Metode *Dijkstra* sesuai *node* asal dan *node* tujuan dari sebuah rute. Untuk mencari rute manakah yang harus dilewati dan memiliki jarak yang paling dekat, maka grafik di atas ditambahkan beberapa kotak untuk mengisi beberapa label. Langkah-langkahnya adalah sebagai berikut :

1. Mengisi kotak label pada titik awal dengan urutan 1 dan label jarak 0.
2. Menetapkan label jarak sementara untuk semua titik yang dapat dihubungi langsung dari titik awal.
3. Pilih titik dengan label jarak sementara terkecil dan menuliskan nilainya di label jarak, serta tambahkan label urutannya.
4. Masukkan label jarak sementara pada setiap titik yang belum memiliki label urutan dan label jarak dan dapat dihubungi langsung dari titik yang baru saja ditulis label jarak dan label urutannya, nilainya diisi dengan total dari label jarak dari titik sebelumnya dan titik tersebut. Jika label sementara di titik tersebut sudah memiliki nilai, maka harus diganti hanya jika nilai yang baru lebih kecil.
5. Pilih titik dengan label jarak sementara terkecil dan menggunakan label jarak semmentarnya sebagai label jarak dari titik tersebut, serta tambahkan label urutannya.
6. Ulangi langkah ke empat dan ke lima hingga titik tujuan memiliki label jarak dan label urutan.
7. Untuk mengetahui rute mana yang harus dilewati adalah dengan menelusuri kembali dari titik tujuan ke titik awal.

Tabel 2 digunakan sebagai acuan dalam penghitungan rute terpendek pada pencarian rute terpendek SLB Kota Medan menggunakan Metode *Dijkstra* sesuai *node* asal dan *node* tujuan dari sebuah rute. Berdasarkan tabel analisis, maka gambar dari graf yang akan diselesaikan *dijkstra* apabila verteks asal ialah A dan verteks tujuan ialah G. Algoritma *Dijkstra* mencari jarak terpendek dari verteks asal ke verteks terdekatnya, kemudian ke verteks berikutnya, dan seterusnya dengan ketentuan :

$$d_{i(\text{baru})} = \min \{ d_{i(\text{lama})}, d_j + m_{ji} \}$$

Maka program akan berhenti karena semua node / verteks sudah terpilih. Dan menghasilkan jalur terpendek dari verteks A ke verteks G, sehingga akan didapat :

1. Rute ke 1
Rute yang dilalui = A→D→E→G
Jarak yang dilalui = 0 + 12 + 3 + 18 = 33
2. Rute ke 2
Rute yang dilalui = A→D→E→I→H→G
Jarak yang dilalui = 0+12+3+13+6+16=50
3. Rute ke 3
Rute yang dilalui = A→D→E→F→H→G
Jarak yang dilalui = 0 + 12 + 3 + 5 + 5 + 16 = 41
4. Rute ke 4
Rute yang dilalui = A→D→F→H→G
Jarak yang dilalui = 0 + 12 + 7 + 5 + 16 = 40
5. Rute ke 5
Rute yang dilalui = A→B→D→E→G
Jarak yang dilalui = 0+6+6+3+18=33
6. Rute ke 6
Rute yang dilalui = A→B→D→E→I→H→G
Jarak yang dilalui = 0+6+6+3+13+6+16=50
7. Rute ke 7
Rute yang dilalui = A→B→D→E→F→H→G
Jarak yang dilalui = 0+6+6+3+5+5+16=41

Sampai Rute Ke 20

$$d_{i(\text{baru})} = \min(33, 50, 41, 40, 33, 50, 41, 40, 40, 61, 74, 56, 54, 51, 62, 60, 58, 44, 42, 39)$$

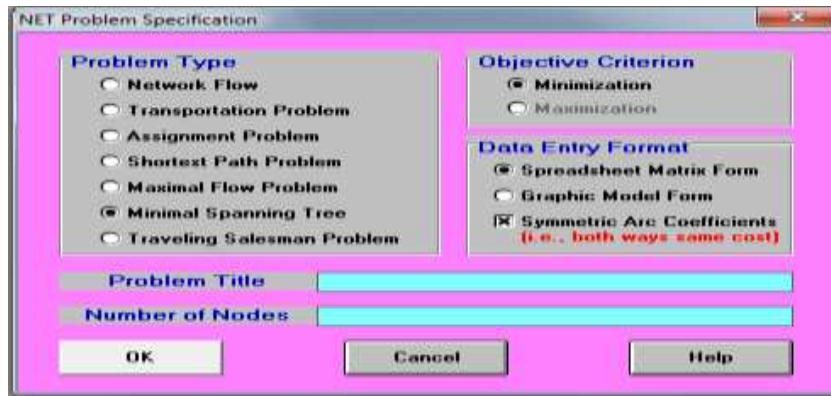
$$d_{i(\text{baru})} = 33$$

Makan rute terpendek verteks A menuju ke verteks G adalah A→D→E→G dan A→B→D→E→G.

4. IMPLEMENTASI

Pada tahap ini akan dilakukan evaluasi terhadap Aplikasi *winqsb* dimana digunakan untuk meminimalisasi biaya pengiriman minuman dalam kemasan. Adapun tahapan yang dilakukan adalah sebagai berikut :

1. Buka aplikasi *WinQsb* dan pilih menu *network modelling*.
2. Klik *File/New* problem untuk membuat proyek baru, pilih *transportation* problem.



Gambar 2. Kotak Dialog Problem Specification

3. Kemudian akan muncul tabel penginputan data seperti pada Gambar 2 dan ketikkan data yang telah didapatkan pada tabel tersebut.

From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10	Node11	Node12	Node13	Node14
Node1		6	6	12									6	
Node2	6			6							7		6	
Node3	6										7		6	
Node4	12	6			3	5								
Node5				3		5	18		13					
Node6			7	5	5									
Node7					18			16				15		
Node8							16							
Node9					13			6						
Node10								17			9		5	7
Node11			7								9			
Node12							15	2						6
Node13	6		6							5	14			
Node14									7			6		

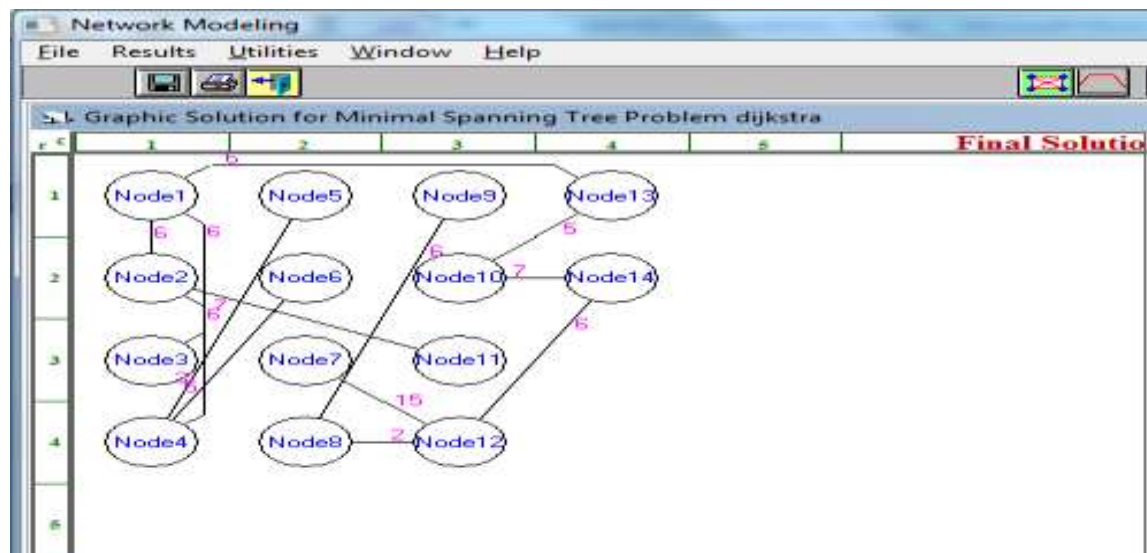
Gambar 3. Tabel peng-input-an data

4. Setelah penginputan data selesai klik solve and analyze/solve the problem sehingga akan muncul solusi yang diperlihatkan oleh Gambar berikut.

09-26-2017	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node1	Node2	6	8	Node9	Node8	6
2	Node1	Node3	6	9	Node13	Node10	5
3	Node2	Node4	6	10	Node2	Node11	7
4	Node4	Node5	3	11	Node14	Node12	6
5	Node4	Node6	5	12	Node1	Node13	6
6	Node12	Node7	15	13	Node10	Node14	7
7	Node12	Node8	2				
	Total	Minimal	Connected	Distance	or Cost	=	80

Gambar 4. Solusi Optimal

Pada Gambar 3 dapat dilihat bahwa solusi optimal dari permasalahan pencarian rute terpendek adalah di atas dapat dilihat pada gambar berikut ini :



Gambar 5. Grafik Solusi Optimal

5. KESIMPULAN

Berdasarkan analisa beberapa hal yang dapat ditarik kesimpulan setelah menulis skripsi ini adalah sebagai berikut:

1. Jumlah node dan edge berpengaruh pada proses waktu untuk mencari solusi dalam pencarian jalur terpendek.
2. Pencarian jalur terpendek dapat menggunakan metode dijkstra sehingga menemukan jarak yang lebih efisien
3. Pertambahan node akan menambah proses waktu yang cukup signifikan dalam pencarian jalur terpendek dibandingkan dengan pertambahan edge..

REFERENCES

- [1] Yogi Primadasa, Jurnal KomTekInfo Fakultas Ilmu Komputer, Volume 2, No. 2, Des 2015.
- [2] Siswanto, 2011, algoritma dan struktur data non linier dengan java, Graha Ilmu Yogyakarta.
- [3] Paryono, P. 1994. Sistem Informasi Geografis. Andi Offset, Yogyakarta.