



SIMULASI JARINGAN JALAN DI KOTA SEMARANG BERBASIS ALGORITMA FLOYD-WARSHALL UNTUK MENANGANI MASALAH LINTASAN TERPENDEK

Harsono✉, Mulyono, Amin Suyitno

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia
Gedung D7 lantai 1 Kampus Sekaran, Gunungpati, Semarang, 50229

Info Artikel

Sejarah Artikel:
Diterima September 2015
Disetujui Oktober 2015
Dipublikasikan Nopember 2016

Keywords:
Simulasi
Algoritma Floyd-Warshall
Lintasan Terpendek

Abstrak

Tujuan penelitian ini untuk mengetahui hasil program simulasi jaringan jalan kota Semarang menggunakan algoritma Floyd-Warshall dengan bahasa pemrograman Visual Basic dan membuktikan bahwa penghitungan manual mempunyai hasil yang sama dengan penghitungan dengan simulasi jaringan jalan kota Semarang dalam mencari lintasan terpendek pada graf. Metode penelitian meliputi (1) Identifikasi dan perumusan masalah, (2) studi pustaka, (3) pengumpulan data, (4) perancangan dan pembuatan program simulasi, (5) implementasi simulasi, (6) evaluasi program simulasi, dan (7) penarikan kesimpulan. Berdasarkan data jaringan jalan kota Semarang yang direpresentasikan ke dalam bentuk graf dan dilakukan simulasi algoritma Floyd-Warshall yang merupakan hasil dari perancangan dan pembuatan program dengan bahasa pemrograman Visual Basic. Simulasi ini dapat menghasilkan lintasan terpendek untuk pencarian dari titik awal ke titik tujuan pada graf yang telah direpresentasikan ke dalam program simulasi. Berdasarkan hasil penghitungan manual dan hasil pengujian program simulasi algoritma Floyd-Warshall pada suatu graf ini terbukti mempunyai solusi yang sama.

Abstract

This study aims to know the result of simulation program of road network by using Floyd-Warshall algorithm and visual basic programming. This study also aims to prove that the manual calculation is not different with the simulation of road networking in finding the shortest path of a graph. The method include (1) problem identification, (2) study of literature, (3) data collection, (4) designing and constructing simulation program, (5) implementation of simulation, (6) evaluation of simulation program, (7) drawing conclusion. Based on road network data of Semarang city which is represented in a form of graph and running simulation Floyd-Warshall algorithm which is the result of designing and creating the program by using visual basic programming. This simulation resulting shortest path for finding from start vertices to destination vertices after represented to simulation program. Based result manual calculation and the simulation Floyd-Warshall algorithm of this graph is not different.

PENDAHULUAN

Teori graf merupakan salah satu cabang matematika yang menarik untuk dibahas karena berkaitan dengan permasalahan yang banyak ditemui dalam kehidupan sehari-hari. Teori graf banyak digunakan untuk mempermudah menyelesaikan suatu masalah. Dengan merepresentasikan persoalan ke dalam bentuk graf, maka persoalan dapat dijelaskan secara lebih sederhana. Banyak sekali aplikasi menggunakan graf sebagai alat untuk mempresentasikan atau memodelkan persoalan sehingga persoalan itu dapat diselesaikan dengan baik. Aplikasi-aplikasi tersebut misalnya menentukan lintasan terpendek (the shortest path problem), persoalan tukang pos, penjadwalan ujian, penentuan frekuensi radio mobile, dan masih banyak lagi (Budayasa, 2007).

Pencarian lintasan terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian lintasan terpendek ini telah diterapkan di berbagai bidang untuk mengoptimasi kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses (Purwananto et al., 2005). Masalah lintasan terpendek secara umum dijelaskan menggunakan konsep graf dapat berupa graf berarah atau graf tidak berarah. Sisi dalam sebuah graf tidak berarah dapat dianggap memungkinkan perjalanan di kedua arah. Sebaliknya, Sisi dalam graf berarah hanya dapat digunakan untuk satu arah perjalanan. Biasanya dalam menentukan lintasan terpendek dengan menggunakan graf berbobot. Setiap sisi dalam graf berbobot terdapat suatu nilai atau bobot (Sushma, 2013).

Sebuah struktur graf dikembangkan dengan memberi bobot pada tiap sisi. Graf berbobot dapat digunakan untuk melambangkan berbagai konsep. Sebagai contoh jika suatu graf melambangkan jaringan jalan maka bobotnya bisa berarti panjang jalan, waktu tempuh maupun batas kecepatan tertinggi jalan tertentu, sehingga untuk menentukan lintasan terpendek diperlukan graf berbobot.

Kesulitan menentukan lintasan terpendek timbul karena terdapat banyak jalan yang ada dari suatu daerah ke daerah lain sehingga memungkinkan memilih jalan alternatif apabila terdapat suatu hambatan pada jalan terpendek utama. Kebutuhan untuk menemukan lintasan terpendek dan waktu tempuh tercepat tentunya juga diperhitungkan

untuk menghindari kerugian seperti contoh bagi sebuah industri. Kebutuhan untuk segera sampai tempat tujuan tepat waktu bahkan diharapkan bisa lebih cepat sangatlah dibutuhkan mengingat persaingan industri yang mementingkan kepuasan pelanggan dan menghindari kerugian karena kerusakan barang. Hal itu dapat saja terjadi bila terjadi pemblokiran jalan secara tiba-tiba pada jalan yang seharusnya dilalui, selain untuk industri lintasan terpendek juga dibutuhkan untuk menghemat waktu tempuh bagi wisatawan yang ingin bepergian ke tempat wisata (Ratnasari et al., 2013). Dalam mencari lintasan terpendek, semakin banyak titik dan garis pada graf akan semakin rumit (Mardlootillah et al., 2014). Salah satu aplikasi pencarian lintasan terpendek yang paling menarik untuk dibahas adalah pada masalah transportasi (Purwananto et al., 2005).

Jaringan transportasi pada kota besar seperti halnya kota Semarang pada umumnya masih mempunyai jaringan yang rumit. Orang akan mengetahui jalan yang harus dilewati untuk sampai ke tempat tujuan yang biasa di kunjunginya, akan tetapi jika tempat tujuan tersebut belum pernah dikunjungi rata-rata mereka sering kesulitan untuk menentukan jalan yang harus dilewati untuk mencapai tempat tersebut. Selama ini orang akan bertanya kepada orang lain yang mengetahui betul jaringan transportasi di kota tersebut. Sehingga lintasan yang mereka lalui menjadi lebih jauh dan dapat membutuhkan lebih banyak waktu serta tenaga yang mereka keluarkan lebih banyak.

Menurut Li, sebagaimana dikutip oleh Xiao-Yan & Yan-Li (2010), bahwa biaya transportasi merupakan komponen yang sangat signifikan. biaya yang dikeluarkan oleh negara berkembang untuk transportasi sebesar 30% dari total biaya ekonomi nasional, sedangkan negara maju hanya 10%. Artinya, hanya dari biaya transportasi, negara berkembang memiliki selisih 20% dari negara maju. Selama dapat menghemat biaya transportasi maka negara berkembang dapat menghemat 10% dari biaya yang ada. Setiap orang dalam melakukan perjalanan pasti memilih jalan terdekat untuk mencapai tempat tujuannya, karena dapat menghemat waktu, tenaga serta biaya. Ada banyak algoritma yang dapat digunakan yang dapat digunakan untuk menentukan lintasan terpendek pada sebuah graf diantaranya Algoritma Djikstra, Algoritma Bellman Ford dan Algoritma Floyd-Warshall.

Dalam penelitian ini peneliti menggunakan Algoritma Floyd-Warshall untuk pencarian lintasan terpendek.

Masalah pencarian lintasan terpendek pada teori graf berkaitan dengan masalah pengoptimuman, antara lain meminimumkan biaya dan efisiensi waktu yang diselesaikan dalam algoritma Floyd Warshall atau algoritma Dijkstra (Syukria et al., 2013). Algoritma Floyd-Warshall merupakan bagian dari program dinamik yang dapat mencari semua lintasan terpendek masing-masing antara tiap kemungkinan pasang tempat yang berbeda (All-pairs Shortest Path Problems) dan sangat efektif digunakan dalam menangani masalah lintasan optimum (Saputra, 2011).

Dari latar belakang tersebut, maka penulis merumuskan beberapa permasalahan yaitu Bagaimana hasil program simulasi jaringan jalan kota Semarang dengan menggunakan algoritma Floyd-Warshall dengan bahasa pemrograman Visual Basic dan Apakah penghitungan manual mempunyai hasil yang sama dengan penghitungan dengan simulasi jaringan jalan kota Semarang dalam mencari lintasan terpendek pada graf. Sejalan dengan rumusan masalah, tujuan penulisan ini adalah untuk mengetahui hasil program simulasi jaringan jalan kota Semarang dengan menggunakan algoritma Floyd-Warshall dengan bahasa pemrograman Visual Basic dan membuktikan bahwa penghitungan manual mempunyai hasil yang sama dengan penghitungan dengan simulasi jaringan jalan kota Semarang dalam mencari lintasan terpendek pada graf.

METODE PENELITIAN

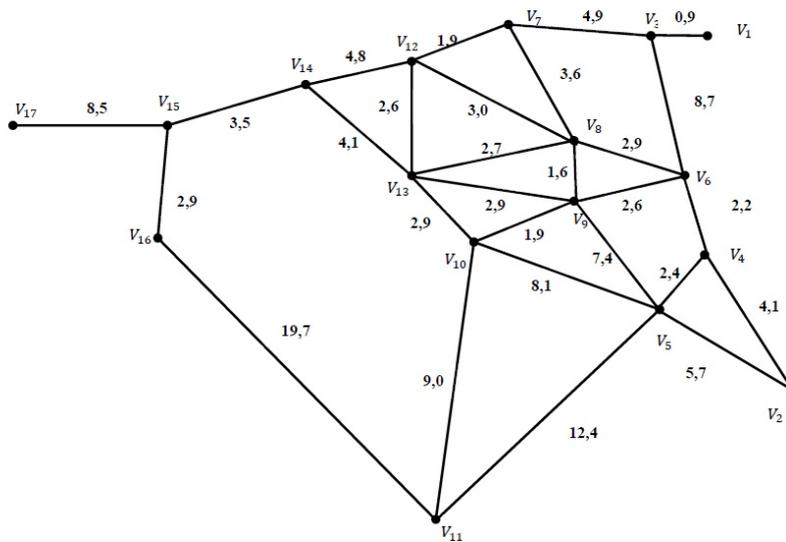
Metode penelitian yang dilakukan dalam penelitian ini meliputi beberapa tahap yaitu identifikasi dan studi pustaka, pengumpulan data, perancangan dan pembuatan program simulasi, implementasi simulasi, evaluasi Program simulasi dan penarikan kesimpulan. Metode yang digunakan dalam pengumpulan data adalah metode observasi, yaitu peneliti mengadakan observasi jaringan jalan yang ada di kota Semarang menggunakan google map dengan tujuan untuk mengetahui jalan-jalan yang ada di kota Semarang. Pada observasi ini peneliti menggunakan google map untuk mengetahui panjang suatu jalan yang dilalui.

HASIL DAN PEMBAHASAN

Tempat-tempat yang akan menjadi titik dalam penelitian ini dibagi berdasarkan kecamatan yang ada di kota Semarang. Tempat-tempat tersebut tidak semua kecamatan yang ada di kota Semarang hanya diambil tempat-tempat yang mewakili kota Semarang. Tempat-tempat tersebut yang akan menjadi titik adalah sebagai berikut.

- a. Terminal Terboyo (v_1)
- b. RSI Sultan Agung (v_3)
- c. Terminal Penggaron (v_2)
- d. RSUD Kota Semarang (v_5)
- e. RSJ Dr. Amino Gondohutomo (v_6)
- f. RS Bhayangkara Semarang (v_4)
- g. Terminal Banyumanik (v_{11})
- h. RS Telogorejo (v_8)
- i. RS Kariadi (v_{13})
- j. RS Roemani Muhammadiyah (v_9)
- k. RS St Elizabeth (v_{10})
- l. Bandara Ahmad Yani (v_{14})
- m. Stasiun Tawang (v_7)
- n. Stasiun Poncol (v_{12})
- o. Terminal Mangkang (v_{17})
- p. RSUD Tugurejo Semarang (v_{15})
- q. RS Permata Medika (v_{16})

Jalan yang digunakan penelitian ini adalah jalan yang bisa kedua arah dan program simulasi yang dibuat adalah program simulasi pencarian lintasan terpendek pada graf tidak berarah. Berdasarkan data peta wilayah kota Semarang dan jarak transportasi yang diperoleh dari google map. Setelah dilakukan penelitian menggunakan google map untuk mengetahui panjang jalan antara 2 tempat. Kemudian disusun gambar jaringan dari data tersebut. Untuk menentukan lintasan terpendek jaringan transportasi kota Semarang, digunakan algoritma Floyd-Warshall. Penggunaan software Visual Basic disini bertujuan untuk mempermudah hasil perhitungan manual dari algoritma Floyd-Warshall karena jaringan yang diperoleh melibatkan titik dan sisi dengan jumlah yang sangat banyak. Dari representasi graf tersebut dapat dicari lintasan terpendek dari tempat awal sampai alamat yang akan dituju. Kemudian dari data tersebut dapat dibuat graf seperti Gambar 1



Gambar 1 Representasi graf jaringan jalan kota Semarang

Setelah merepresentasikan data ke dalam bentuk graf langkah selanjutnya adalah proses manual pencarian lintasan terpendek dengan Algoritma Floyd-Warshall

Diketahui n titik dalam graf tidak berarah adalah $v_1, v_2, v_3, \dots, v_n$ untuk menentukan lintasan terpendek diantara semua pasangan titik, dengan langkah sebagai berikut:

Langkah 1: untuk i, j , jika v_i, v_j adalah sisi, ambil $d(i, j)$ sebagai bobot dari sisi tersebut. Jika tidak ada sisi yang menghubungkan langsung antara i dan j ditulis $d(i, j) = \infty$. Untuk $i=j$, maka ditulis $d(i, j) = 0$.

Langkah 2: untuk $k=1$ sampai n
 untuk $j=1$ sampai n
 untuk $i=1$ sampai n

Ditulis $d((i, j)) = \min\{d(i, j), d(i, k) + d(k, j)\}$

Nilai akhir dari $d(i, j)$ adalah jarak dari v_i ke v_j . (Goodaire & Parmeter, 1998)

Dari prosedur di atas dapat dilihat bahwa pada iterasi ke- k ($1 \leq k \leq n$). Langkah-langkah untuk menentukan lintasan terpendek dengan menggunakan algoritma Floyd-Warshall adalah sebagai berikut.

Langkah pertama adalah mengubah graf ke dalam bentuk matriks.

Setelah diperoleh data yang titik dan jarak antara titik-titik yang ada di kota Semarang. Langkah selanjutnya adalah mengubah data tersebut ke dalam bentuk matriks. Bentuk matriksnya adalah sebagai berikut.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}
v_1	0	∞	∞	0,9	∞												
v_2	∞	0	∞	4,1	5,7	∞											
v_3	0,9	∞	0	∞	∞	8,7	4,9	∞									
v_4	∞	4,1	∞	0	2,4	2,2	∞										
v_5	∞	5,7	∞	2,4	0	∞	∞	∞	7,4	8,1	12,4	∞	∞	∞	∞	∞	∞
v_6	∞	∞	8,7	2,2	∞	0	∞	2,9	2,6	∞							
v_7	∞	∞	∞	4,9	∞	∞	0	3,6	∞	∞	∞	1,9	∞	∞	∞	∞	∞
v_8	∞	∞	∞	∞	∞	2,9	3,6	0	1,9	∞	∞	3,0	2,7	∞	∞	∞	∞
v_9	∞	∞	∞	∞	7,4	2,6	∞	1,6	0	1,9	∞	∞	2,9	∞	∞	∞	∞
v_{10}	∞	∞	∞	∞	∞	8,1	∞	∞	1,9	0	9,0	∞	2,9	∞	∞	∞	∞
v_{11}	∞	∞	∞	∞	∞	∞	12,4	∞	∞	∞	9,0	0	∞	∞	∞	∞	19,7
v_{12}	∞	1,9	3,0	∞	∞	0	2,6	4,8	∞	∞							
v_{13}	∞	2,7	2,9	2,9	∞	2,6	0	4,1	∞	∞							
v_{14}	∞	4,8	4,1	0	3,5												
v_{15}	∞	3,5	0	2,9													
v_{16}	∞	2,9	0														
v_{17}	∞	8,5	∞														

Langkah selanjutnya setelah mengubah data kedalam bentuk matriks adalah melakukan iterasi Untuk $k=1$ sampai 17. Setelah dilakukan pencarian semua nilai $d(i, j)$ untuk setiap i dan j dari untuk $k=1$ sampai $k = 17$. Diperoleh sebuah matriks dibawah ini.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}
v_1	0	15,9	0,9	11,8	14,2	9,6	5,8	9,4	11	12,9	21,9	7,7	10,3	12,5	16	18,9	24,5
v_2	15,9	0	15	4,1	5,7	6,3	12,8	9,2	8,9	10,8	18,1	12,2	11,8	15,9	19,4	22,3	27,9
v_3	0,9	15	0	10,9	13,3	8,7	4,9	8,5	10,1	12	21	6,8	9,4	11,6	15,1	18	23,6
v_4	11,8	4,1	10,9	0	2,4	2,2	8,7	5,1	4,8	6,7	14,8	8,1	7,7	11,8	15,3	18,2	23,8
v_5	14,2	5,7	13,3	2,4	0	4,6	11,1	7,5	7,2	8,1	12,4	10,5	10,1	14,2	17,7	20,6	26,2
v_6	9,6	6,3	8,7	2,2	4,6	0	6,5	2,9	2,6	4,5	13,5	5,9	5,5	9,6	13,1	16	21,6
v_7	5,8	19,9	4,9	8,7	11,1	6,5	0	3,6	5,2	7,1	16,1	1,9	4,5	6,7	10,2	13,1	18,7
v_8	9,4	9,2	8,5	5,1	7,5	2,9	3,6	0	1,6	3,5	12,5	3,0	2,7	6,8	10,3	13,2	18,8
v_9	11	8,9	10,1	4,8	7,2	2,6	5,2	1,6	0	1,9	10,9	4,6	2,9	7	10,5	13,4	19,1
v_{10}	12,9	10,8	12	6,7	8,1	4,5	7,1	3,5	1,9	0	9,0	5,5	2,9	7	6,4	9,3	14,9
v_{11}	21,9	18,1	21	14,8	12,4	13,5	16,1	12,5	10,9	9,0	0	14,5	11,9	16	19,5	19,7	28
v_{12}	7,7	12,2	6,8	8,1	10,5	5,9	1,9	3,0	4,6	5,5	14,5	0	2,6	4,8	6,1	9	14,6
v_{13}	10,3	11,8	9,4	7,7	10,1	5,5	4,5	2,7	2,9	2,9	11,9	2,6	0	4,1	7,6	10,5	16,1
v_{14}	12,5	15,9	11,6	11,8	14,2	9,6	6,7	6,8	7	7	16	4,8	4,1	0	3,5	6,4	12
v_{15}	16	19,4	15,1	15,3	17,7	13,1	10,2	10,3	10,5	6,4	19,5	6,1	7,6	3,5	0	2,9	8,5
v_{16}	18,9	22,3	18	18,2	20,6	16	13,2	13,2	13,4	9,3	19,7	9	10,5	6,4	2,9	0	11,4
v_{17}	24,5	27,9	23,6	23,8	26,2	21,6	18,7	18,8	19	14,9	28	14,6	16,1	12	8,5	11,4	0

Berdasarkan matriks di atas menunjukkan bahwa jarak-jarak untuk setiap titik telah ditemukan. matriks di atas menunjukkan jarak atau $d(i, j)$ karena $k = n$. Untuk menentukan lintasan terpendek dapat di

cari dari matriks $d(i,j)$. berdasarkan matriks di atas diperoleh jarak dari v_1 ke v_2 adalah 15,9, v_1 ke v_3 adalah 0,9, v_1 ke v_4 adalah 11,8, v_1 ke v_5 adalah 12,6 dan seterusnya.

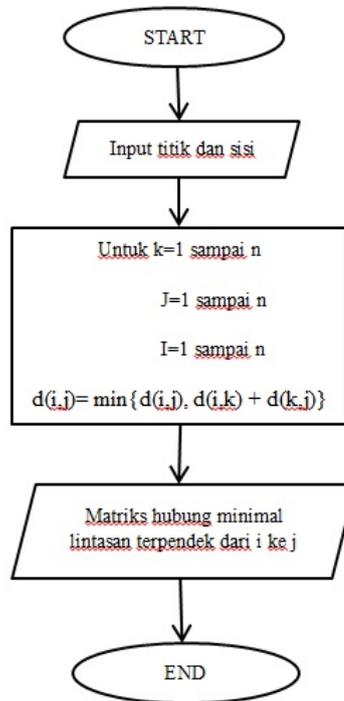
Setelah dilakukannya analisis manual data langkah selanjutnya adalah tahap perancangan simulasi dan pembuatan program simulasi. Pada tahap perancangan sistem Simulasi ini, akan dilakukan penggambaran tampilan-tampilan yang akan dibangun. Tampilan tersebut berupa tampilan input, tampilan proses dan tampilan Output. Output merupakan tampilan hasil pengolahan data yang dimasukkan ke dalam suatu sistem dan dilakukan suatu proses sesuai dengan algoritma yang telah ditentukan. Permasalahan yang diselesaikan dalam sistem ini adalah bagaimana menentukan lintasan terpendek dan dilanjutkan dengan penentuan jaraknya. Gambar 2 merupakan flowchart dari sistem.



Gambar 2 Rancangan simulasi

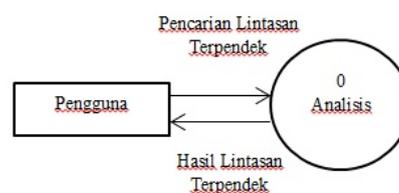
Pencarian lintasan terpendek menggunakan algoritma Floyd-Warshall dapat digambarkan menggunakan flowchart. Flowchart yang dibuat bertujuan untuk mempermudah pembuatan source code program simulasi dalam menangani masalah lintasan terpendek pada graf yang dibangun menggunakan bahasa pemrograman Visual Basic 6.0. Flowchart juga akan membuat logika berfikir pembuatan source code program simulasi ini tersusun dengan baik

berdasarkan urutan (step) yang sesuai dengan algoritma Floyd-Warshall yang dipakai. Flowchart algoritma Floyd-Warshall dalam menangani masalah pencarian lintasan terpendek pada graf dapat digambarkan seperti pada Gambar 3.



Gambar 3 Flowchart algoritma Floyd-Warshall

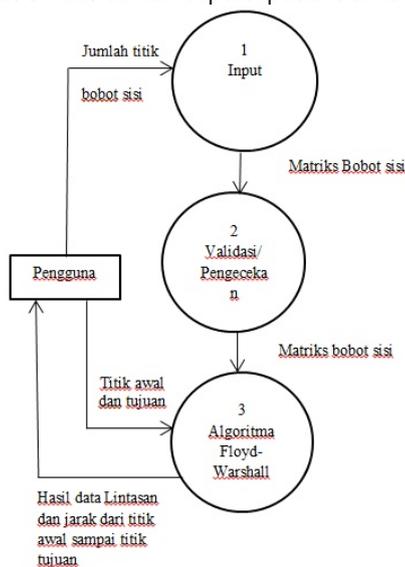
Kebutuhan data maupun fungsional yang diperlukan oleh sistem digambarkan dalam diagram alir data. Secara keseluruhan sistem ini memiliki beberapa level dalam pemodelan diagram, diantaranya : Context Diagram atau DFD level 0, dan DFD level 1. Context Diagram sebagai top diagram yang dapat memberikan gambaran umum dari sebuah sistem, diantaranya: pemodelan aliran-aliran masuk dan keluar baik yang berasal dari sistem maupun yang berasal dari entitas-entitas eksternal. Sedangkan masing-masing level pada DFD menggambarkan proses yang lebih rinci terhadap perubahan data dan fungsionalitas dalam sistem.



Gambar 4 DFD level 0

Gambar 4 merupakan context diagram atau yang disebut dengan Data Flow Diagram (DFD) Level 0 merupakan alat yang digunakan untuk mendokumentasikan proses dalam sistem. Tujuannya adalah memberikan gambaran proses dalam sistem secara umum. Pengguna adalah orang yang menjalankan program simulasi untuk menemukan pemecahan masalah lintasan terpendek dari kasus yang diinginkan menggunakan algoritma Floyd-Warshall.

Tahap lanjutan adalah membuat DFD Level 1. Pembuatan DFD Level 1 ini bertujuan untuk lebih memperjelas tahapan jalannya program yang digunakan oleh pengguna (user). Gambar DFD Level 1 menjelaskan proses penggunaan simulasi dimulai dari penginputan data dalam bentuk representasi graf sampai diperoleh hasil lintasan terpendek dari titik awal menuju titik tujuan yang diinginkan. Proses tersebut diilustrasikan seperti pada Gambar 5.

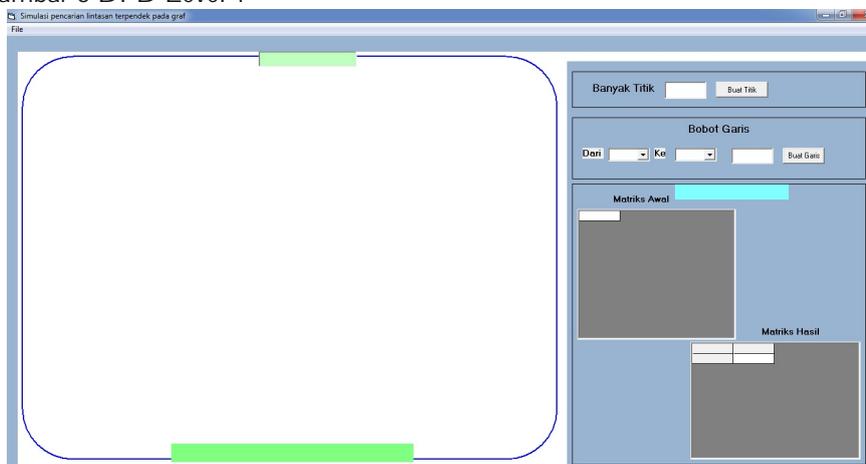


Gambar 5 DFD Level 1

Pengguna(User) menginput data pada yang berupa titik dan bobotnya. Sekaligus input untuk memeriksa program yaitu titik awal dan titik tujuan. Setelah data diinputkan semua, maka akan diperoleh sebuah matriks dari data-data yang telah diinputkan. Matriks yang diperoleh selanjutnya akan dilakukan sebuah pengecekan. Program dapat bekerja sesuai dengan titik awal dan titik tujuan yang diinputkan pengguna. Sehingga akan dihasilkan output berupa lintasan terpendek dari titik awal ke titik tujuan berdasarkan yang diinputkan pengguna.

Program yang telah dirancang dengan desain program agar dapat menjelaskan dan mempermudah dalam merancang program yang akan dibuat. Dari desain program kemudian diterjemahkan ke dalam kata-kata bahasa pemrograman dengan menggunakan bahasa pemrograman Microsoft Visual Basic 6.0 agar komputer dapat melakukannya sesuai dengan keinginan. Sedangkan tahap akhirnya yaitu tahap penggunaan dan aplikasi mulai dioperasikan untuk melakukan berbagai penghitungan pencarian lintasan terpendek pada berbagai kasus yang dapat digambarkan ke dalam graf.

Proses input graf dilakukan dengan cara menggambar titik dan sisi yang menghubungkan setiap titik pada work area. Selanjutnya adalah memberi nama titik tersebut dan memberi bobot pada sisi antara titik yang satu dengan yang lainnya. Adapun bentuk halaman untuk proses penginputan graf seperti pada Gambar 6

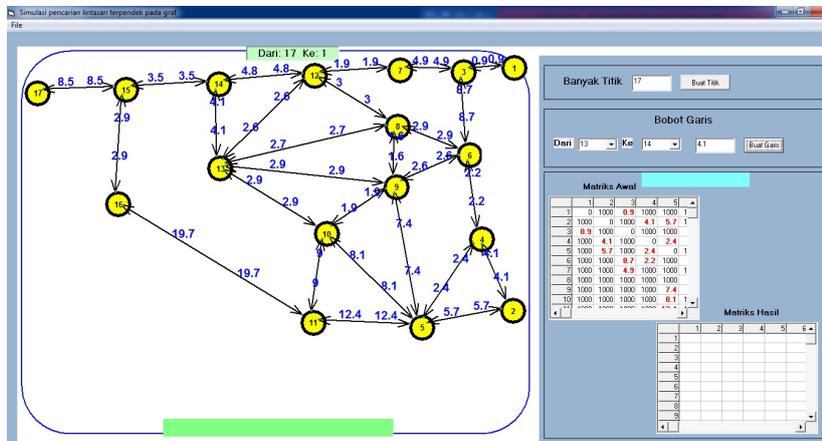


Gambar 6 Tampilan awal program

Gambar 6 merupakan gambar tampilan utama program. Tampilan program dirancang dan direalisasikan agar mudah digunakan oleh pemakai yang bersifat user friendly. Pemakai dapat melakukan simulasi pencarian lintasan berupa rute perjalanan dari titik asal ke titik yang dituju dan menghasilkan keluaran berupa suatu informasi titik-titik yang harus dilewati dan jaraknya.

Setelah tahap perancangan simulasi dan pembuatan simulasi, langkah selanjutnya adalah tahap implementasi simulasi. Tahap implementasi simulasi merupakan tahap untuk pengujian program setelah dilakukannya

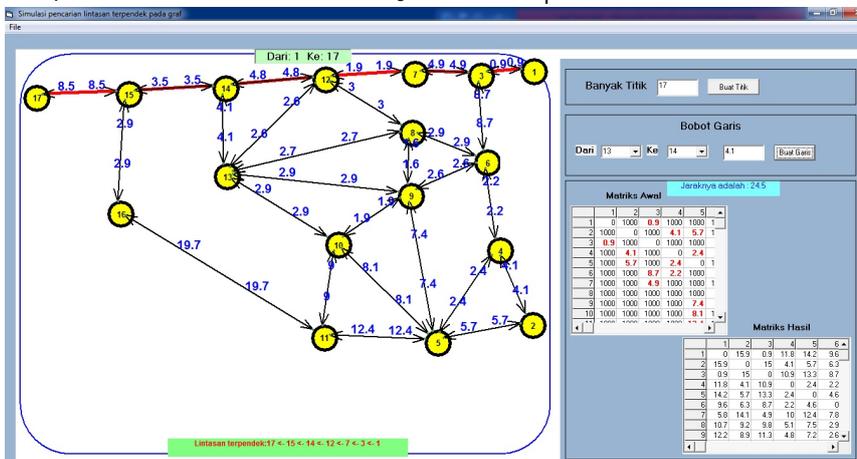
perancangan dan pembuatan program. Pengujian program digunakan untuk melihat hasil dari program simulasi pencarian lintasan terpendek yang telah dibuat. Graf yang direpresentasikan dalam program simulasi adalah berdasarkan data jaringan jalan kota Semarang. Setelah graf dibentuk sesuai dengan data yang diperoleh maka langkah selanjutnya adalah menentukan titik awal dan titik tujuan. Setelah data-data dimasukkan maka akan diperoleh sebuah matriks. Berikut ini merupakan representasi graf dapat dilihat pada Gambar 7.



Gambar 7 Representasi graf pada simulasi

Gambar 7 penentuan titik awal dan titik tujuan dilakukan setelah bobot dari graf yang direpresentasikan masuk ke dalam program simulasi dengan cara mengklik titik yang ditetapkan sebagai titik awal dan titik tujuan. Kemudian dapat dilakukan analisis Floyd-

Warshall dengan cara mengklik tombol analisis untuk memperoleh lintasan terpendek dari graf yang telah direpresentasikan tersebut. Sehingga diperoleh lintasan terpendek dari titik awal ke titik tujuan. Hasil program simulasi dapat dilihat pada Gambar 8.



Gambar 8. Hasil Program Simulasi

Gambar 8 merupakan hasil pencarian lintasan terpendek dari titik v_1 ke titik v_{17} dengan menggunakan simulasi algoritma Floyd-Warshall menggunakan bahasa pemrograman Visual Basic ditandai dengan garis berwarna merah dan tebal yaitu $v_1-v_3-v_7-v_{12}-v_{14}-v_{15}-v_{17}$ dengan jarak adalah 24,5. Dengan demikian, jika dilakukan perbandingan antara hasil solusi pencarian lintasan terpendek menggunakan simulasi dengan cara manual ternyata mempunyai hasil yang sama. Jadi, simulasi algoritma Floyd-Warshall dalam menangani masalah lintasan terpendek pada graf dapat diimplementasikan dengan berbagai macam bentuk graf yang dapat direpresentasikan ke dalam simulasi.

PENUTUP

Berdasarkan hasil penghitungan manual dan hasil pengujian program simulasi algoritma Floyd-Warshall untuk menangani lintasan terpendek pada suatu graf ini terbukti mempunyai solusi yang sama. Berdasarkan data jaringan jalan di kota Semarang yang direpresentasikan ke dalam bentuk graf dan dilakukan simulasi algoritma Floyd-Warshall untuk menangani masalah pencarian lintasan terpendek. Simulasi ini merupakan hasil dari perancangan dan pembuatan program dengan bahasa pemrograman Visual Basic. Simulasi ini dapat menghasilkan lintasan terpendek dan jarak untuk pencarian dari titik awal ke titik tujuan pada graf yang telah direpresentasikan ke dalam program simulasi.

Daftar Pustaka

- Budayasa, I K. 2007. Teori Graf dan Aplikasinya. Surabaya: Unesa University Press.
- Goodaire, E. G dan Parmenter, M. M. 1998. Discrete Mathematics with graf theory. Presenticehall, USA
- Mardlootillah, H. I., Suyitno. A & Arini, F. Y. 2014. Simulasi Algoritma Djikstra dalam Menangani Masalah Lintasan Terpendek [ada Graf Menggunakan Visual Basic. UNNES Journal of

- Mathematics, 3(1): 56-61.
- Purwananto, Y., Purwitasari, A., & Wibowo A. W. 2005. Implementasi dan Analisis Algoritma Pencarian Rute Terpendek di Kota Surabaya. Jurnal Penelitian dan Pengembangan TELEKOMUNIKASI, 10(2): 94-101
- Ratnasari, A., Ardiani, F. & Nurvita, F. A. 2013. Penentuan Jarak Terpendek dan Jarak Terpendek Alternative Menggunakan Algoritma Djikstra Serta Estimasi Waktu Tempuh. Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013. Universitas Islam Indonesia.
- Saputra, R. 2011. Sistem Informasi Geografis pencarian Rute Optimum Objek Wisata Kota Yogyakarta dengan Algoritma Floyd-Warshall. Jurnal Matematika, 14(1): 19-24.
- Sushma, J. P. 2013. Shortest Path Algorithms Techniques. International Journal of Science and Modern Engineering. 1(10):8-12.
- Syukria, A., Johar, R., & Marwan. 2013. Kemampuan Komunikasi Matematis dan Habits of Mind Mahasiswa Pada Materi Lintasan Terpendek Menggunakan Algoritma Floyd-Warshall. Jurnal Peluang, 1(2):71-80.
- Xiao-Yan, L. & Yan-Li, C. 2010. Application of Djikstra Algorithm in Logistics Distribution Lines. Proceedings 3rd International Symposium on Computer Science and Computational Technology. Henan Polytechnic University.