

APLIKASI ALGORITMA PRIM UNTUK MENENTUKAN MINIMUM SPANNING TREE SUATU GRAF BERBOBOT DENGAN MENGGUNAKAN PEMROGRAMAN BERORIENTASI OBJEK

Deny Wiria Nugraha

Dosen Jurusan Teknik Elektro UNTAD Palu, Indonesia

email : deny_wiria_nugraha@yahoo.co.id

Abstract- In this study, the algorithm used is Prim's algorithm—an algorithm in graph theory to seek a minimum spanning tree for a weighted connected graph. The program used is a program created with Delphi 7 programming language used for searching the minimum spanning tree of a graph model with the weight of each form of distance/length connecting the points/vertices. Then display the information search process sequence of minimum spanning tree, the total number of minimum length and the resulting computing time to determine the efficiency of Prim's algorithm. Based on the results of research, Prim's algorithm computation time in finding the minimum spanning tree of a weighted graph will grow up along with increasing the number of points/vertices and the number of sides of the weighted graph.

Keywords: Prim's Algorithm, Graph model, Minimum Spanning Tree

I. PENDAHULUAN

Banyak permasalahan yang dapat dimodelkan dengan menggunakan graf, khususnya di bidang teknik elektro dan teknologi informasi. Salah satunya adalah masalah dalam pencarian pohon merentang minimum (*minimum spanning tree*). Graf mempunyai peruntukan yang cukup luas dalam kehidupan nyata, diantaranya adalah penggunaan graf dalam melakukan perutean. Perutean yaitu kegiatan membuat rute atau jalur dengan tujuan tertentu. Dengan penggunaan graf akan didapatkan lintasan dengan keunggulan-keunggulan tertentu misalnya lintasan dengan biaya paling murah,

lintasan dengan waktu tempuh paling cepat, lintasan dengan jarak paling pendek, dan lintasan dengan tingkat efisiensi paling tinggi. Pada bidang yang lebih luas, jika dianalogikan lintasan yang dibuat dengan alur kerja yang harus dilakukan, maka akan didapatkan efisiensi kerja dan hasil yang optimal.

Di antara sekian banyak konsep dalam teori graf, konsep pohon merupakan konsep yang paling penting, karena terapannya yang luas dalam berbagai bidang ilmu. Banyak terapan, baik dalam bidang ilmu komputer maupun di luar bidang ilmu komputer, yang telah mengkaji pohon secara intensif sebagai objek matematika. Dalam kehidupan sehari-hari, pohon banyak digunakan untuk menggambarkan hirarkhi. Misalnya pohon silsilah keluarga, struktur organisasi, organisasi pertandingan, dan lain-lain.

Hasil telaah literatur mengindikasikan bahwa penelitian tentang penggunaan suatu algoritma untuk menentukan pohon merentang minimum dan implementasinya pada suatu graf berbobot pernah dilakukan oleh sejumlah peneliti, antara lain: Gloor, *et al.* (1993) melakukan pembuktian kebenaran suatu algoritma dengan melakukan visualisasi. Greenberg (1998) membandingkan algoritma Prim dan algoritma Kruskal dalam mencari pohon merentang minimum dengan menggunakan graf yang terhubung dengan bobot tidak negatif pada sisi-sisinya. Pop dan Zelina (2004) melakukan penelitian dengan menyajikan algoritma waktu eksponensial untuk graf tidak berarah yang memiliki simpul-simpul n yaitu algoritma heuristik berbasis Kruskal, algoritma heuristik berbasis Prim, dan algoritma heuristik berbasis pendekatan global-lokal.

Pada prinsipnya penelitian ini bertujuan untuk membuktikan/menguji bahwa algoritma Prim dapat digunakan untuk menentukan pohon merentang minimum suatu graf berbobot dan mengetahui unjuk-kerja aplikasi algoritma Prim berupa waktu komputasi dalam menentukan pohon merentang minimum suatu graf berbobot.

Penelitian ini diharapkan dapat memberikan kontribusi terhadap dunia akademisi yaitu untuk menambah wawasan dan ilmu pengetahuan tentang algoritma, khususnya mengenai graf, graf berbobot, pohon merentang minimum dan waktu komputasi algoritma Prim. Penelitian ini diharapkan juga dapat memberikan masukan kepada praktisi dalam melakukan perancangan suatu model graf berbobot, sehingga dapat memudahkan melakukan perhitungan dan monitoring terhadap suatu sistem yang dapat membantu memecahkan permasalahan yang ada hubungannya dengan penentuan rute atau jalur dengan menggunakan pohon merentang minimum.

Sesuai dengan ilustrasi yang diuraikan pada pendahuluan, maka rumusan masalah dalam penelitian ini adalah bagaimanakah algoritma Prim diaplikasikan untuk menentukan pohon merentang minimum pada suatu graf berbobot dan bagaimanakah program bantu yang dibuat dalam penelitian ini mampu memodelkan graf dan menghitung data jarak/panjang dengan menggunakan metode algoritma Prim.

II. Tinjauan Pustaka

A. Algoritma

Algoritma adalah deskripsi langkah-langkah penyelesaian masalah yang tersusun secara logis atau urutan logis pengambilan keputusan untuk pemecahan suatu masalah. Algoritma ditulis dengan notasi khusus, notasi mudah dimengerti dan notasi dapat diterjemahkan menjadi sintaks suatu bahasa pemrograman (Zakaria dan Prijono, 2006).

Algoritma merupakan salah satu cabang ilmu komputer yang membahas prosedur penyelesaian suatu permasalahan. Dengan algoritma yang baik maka komputer bisa menyelesaikan perhitungan dengan cepat dan benar. Sebaliknya jika algoritma kurang

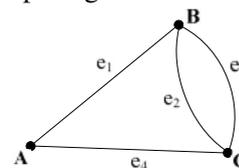
baik maka penyelesaian lambat dan bahkan tidak didapat solusi yang diharapkan.

Suatu algoritma akan memerlukan masukan (input) tertentu untuk memulainya, dan akan menghasilkan keluaran (output) tertentu pada akhirnya. Hal-hal yang perlu diperhatikan dalam algoritma adalah mencari langkah-langkah yang paling sesuai untuk penyelesaian suatu masalah, karena setiap algoritma memiliki karakteristik tertentu yang memiliki kelebihan dan kekurangan. Beberapa hal yang harus dipahami dalam mencari algoritma antara lain:

1. Masalah seperti apa yang hendak diselesaikan.
2. Gagasan apa yang ada pada algoritma tersebut.
3. Berapa lama yang diperlukan untuk menyelesaikan masalah.
4. Berapa jumlah data yang dapat ditangani oleh suatu algoritma.

B. Graf (Graph)

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$. Dalam hal ini, V merupakan himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) digambarkan dalam titik-titik, dan E adalah himpunan sisi-sisi (*edges* atau *arcs*) digambarkan dalam garis-garis yang menghubungkan sepasang simpul (Munir, 2009). Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi. Graf dapat digambarkan pada gambar 1.



Gambar 1. Graf G

Pada gambar graf G diatas, graf terdiri dari himpunan V dan E yaitu:

$$V = (A, B, C) \dots\dots\dots (1)$$

$$E = (e_1, e_2, e_3, e_4); \text{ bisa ditulis } \{(A,B),(B,C),(B,C),(A,C)\} \dots\dots (2)$$

C. Graf Terhubung (*Connected Graph*)

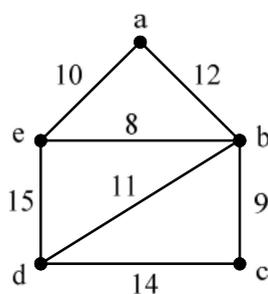
Graf G disebut graf terhubung jika untuk setiap pasang simpul u dan v di dalam himpunan V terdapat lintasan dari u ke v . Jika tidak, maka graf G disebut graf tak-terhubung (*disconnected graph*) (Munir, 2009).

Keterhubungan dua buah simpul adalah penting di dalam graf. Jika dua buah simpul terhubung maka pasti simpul yang pertama dapat dicapai dari simpul yang kedua.

Pada graf berarah, graf G dikatakan terhubung jika graf tak-berarahnya terhubung (graf tak-berarah dari G diperoleh dengan menghilangkan arahnya). Keterhubungan dua buah simpul pada graf berarah dibedakan menjadi terhubung kuat dan terhubung lemah. Dua simpul, u dan v pada graf berarah G disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari u ke v dan juga sebaliknya lintasan berarah dari v ke u . Jika u dan v tidak terhubung kuat tetapi tetap terhubung pada graf tak-berarahnya, maka u dan v dikatakan terhubung lemah (*weakly connected*).

D. Graf Berbobot (*Weighted Graph*)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga. Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah tiang listrik, kapasitas, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain, ongkos produksi, dan sebagainya. Untuk lebih jelasnya, graf berbobot dapat digambarkan pada gambar 2.



Gambar 2. Graf berbobot

E. Matriks Ketetanggaan (*Adjacency Matrix*)

Matriks ketetanggaan atau matriks kedekatan adalah representasi graf yang paling umum. Matriks ketetanggaan graf G adalah matriks yang berukuran $n \times n$. Bila matriks tersebut dinamakan $A = [a_{ij}]$, maka $a_{ij} = 1$ jika simpul i dan j bertetangga, sebaliknya $a_{ij} = 0$ jika simpul i dan j tidak bertetangga (Zakaria dan Prijono, 2006).

Jumlah elemen matriks ketetanggaan untuk graf dengan n simpul adalah n^2 . Jika tiap elemen membutuhkan ruang memori sebesar p , maka ruang memori yang diperlukan seluruhnya adalah pn^2 .

Keuntungan representasi dengan matriks ketetanggaan adalah elemen matriksnya dapat diakses langsung melalui indeks. Selain itu juga dapat menentukan dengan langsung apakah simpul i dan j bertetangga.

Untuk graf berbobot, a_{ij} menyatakan bobot tiap sisi yang menghubungkan simpul i dengan simpul j . Pada gambar 2 jika dibuat representasi dalam matriks ketetanggaan maka akan menghasilkan tabel 1.

Tabel 1. Matriks ketetanggaan graf berbobot

	a	b	c	d	e
a	0	12	∞	∞	10
b	12	0	9	11	8
c	∞	9	0	14	∞
d	∞	11	14	0	15
e	10	8	∞	15	0

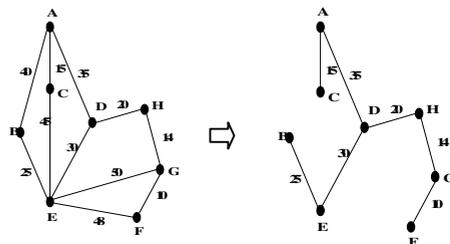
Tanda " ∞ " menyatakan bahwa tidak ada sisi dari simpul i ke simpul j sehingga a_{ij} dapat diberi nilai tak berhingga dan angka "0" menyatakan bahwa simpul saling berhubungan dengan simpulnya sendiri.

F. Pohon Merentang Minimum (*Minimum Spanning Tree*)

Apabila G adalah graf berbobot, maka bobot pohon merentang T dari G didefinisikan sebagai jumlah bobot semua sisi di T . Pohon merentang yang berbeda mempunyai bobot yang berbeda pula. Diantara semua pohon merentang dalam graf G , pohon merentang yang berbobot minimum dinamakan pohon merentang minimum. Pohon merentang minimum ini mempunyai terapan yang luas dalam masalah riil (Munir, 2009).

Misalkan terdapat graf yang sangat kompleks, dimana ada beberapa alternatif untuk melakukan kunjungan-kunjungan (*visiting*) dari satu simpul ke simpul-simpul yang lainnya, tentu dapat segera dicari tahu alternatif yang terbaik untuk melakukannya. Dalam hal ini, alternatif yang cukup baik adalah dengan mencari jarak yang terdekat antar simpul itu.

Contoh: Misalkan akan dibangun jaringan distribusi listrik primer yang menghubungkan sejumlah titik tiang di suatu daerah, dalam rancangannya digambarkan pada gambar 3.



Gambar 3. Contoh graf berbobot rancangan jaringan distribusi listrik primer dan pohon merentang minimum yang terbentuk

Langkah-langkah menghitung total jarak minimum dari suatu graf sebagai berikut:

1. Dari suatu graf yang terbentuk, perhatikan apakah memenuhi kriteria suatu pohon merentang.
2. Lakukan pelacakan secara berurutan mulai dari simpul pertama sampai dengan simpul terakhir.
3. Pada setiap simpulnya perhatikan nilai (bobot) tiap-tiap sisinya.
4. Ambil nilai yang paling kecil artinya jarak terpendek dari setiap sisi simpul.
5. Lanjutkan sampai seluruh simpul tergambar pada pohon merentang.
6. Jumlahkan nilai yang telah dipilih atau jarak minimum yang menghubungkan simpul-simpul tersebut.

G. Algoritma Prim

Algoritma Prim dimulai dari simpul yang berubah-ubah di setiap tingkatnya, diperbolehkan menambah cabang baru untuk membuat susunan pohon baru. Algoritma ini akan tertahan ketika simpul yang sedang dieksplorasi pada graf sudah sampai pada

simpul yang dituju. Strategi yang digunakan adalah strategi Greedy dengan menganggap bahwa pada setiap langkah dari pohon merentang adalah *augmented* dan dipilih simpul yang nilainya paling kecil dari semua simpul yang ada (Purwanto, 2008).

Algoritma Prim menitikberatkan pada pemilihan bobot minimum berdasarkan simpul yang diambil. Dan karena tidak perlu mengurutkan terlebih dahulu, algoritma Prim cocok untuk pohon dengan jumlah simpul banyak. Algoritma Prim akan selalu berhasil menemukan pohon merentang minimum tetapi pohon merentang yang dihasilkan tidak selalu unik.

Langkah-langkah dalam algoritma Prim adalah sebagai berikut:

1. Buat sebuah pohon yang terdiri dari satu simpul (*node*), dipilih secara acak dari graf.
2. Buat sebuah himpunan yang berisi semua cabang di graf.
3. Loop sampai semua cabang di dalam himpunan menghubungkan dua simpul di pohon
 - a. Hapus dari himpunan satu cabang dengan bobot terkecil yang menghubungkan satu simpul di pohon dengan satu simpul di luar pohon.
 - b. Hubungkan cabang tersebut ke pohon.

Algoritma Prim dalam *pseudocode* yaitu:

Procedure Prim(input G:graf, output T:pohon)

{ Membentuk pohon merentang minimum T dari graf terhubung G.

Masukan: graf-berbobot terhubung $G = (V, E)$, yang mana $|V| = n$

Keluaran: pohon merentang minimum $T = (V, E')$ }

Deklarasi

i, p, q, u, v : integer

Algoritma

Cari sisi (p,q) dari E yang berbobot terkecil

$T \leftarrow \{(p,q)\}$

for $i \leftarrow 1$ to $n-1$ do

Pilih sisi (u,v) dari E yang bobotnya terkecil namun bersisian dengan suatu simpul didalam T

$T \leftarrow T \cup \{(u,v)\}$

Endfor

Keterangan :

G = graf.

T = pohon.

V = himpunan simpul-simpul.

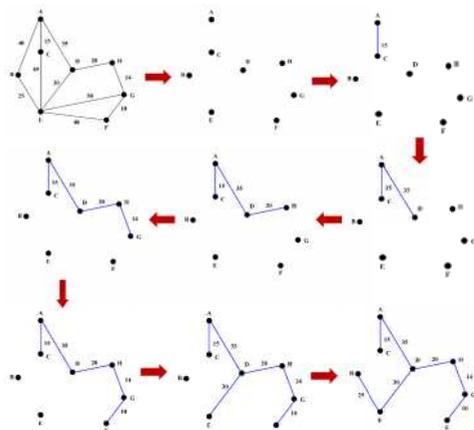
E = himpunan sisi-sisi yang menghubungkan simpul di graf.

E' = himpunan sisi-sisi yang menghubungkan simpul-simpul yang ada di pohon.

(p,q) = pasangan simpul-simpul yang ada di graf yang membentuk sisi.

(u,v) = pasangan simpul-simpul yang ada di pohon yang membentuk sisi dengan bobot terkecil.

Pembentukan/penentuan pohon merentang minimum dengan algoritma Prim pada suatu graf berbobot dapat dilihat pada gambar 4.



Gambar 4. Pembentukan pohon merentang minimum dengan algoritma Prim

III. Metode Penelitian

A. Bahan Penelitian

Data yang merupakan bahan penelitian ini dikumpulkan melalui beberapa metode sebagai berikut:

1. Studi literatur, yaitu penelusuran literatur mengenai dasar pengetahuan tentang hal-hal yang berkaitan dengan penelitian ini. Metode ini dilakukan dengan cara mencari buku-buku, artikel-artikel, dan jurnal-jurnal ilmiah mengenai algoritma khususnya mengenai graf, pohon merentang minimum dan algoritma Prim.
2. Pengumpulan data berupa data panjang atau jarak yang akan digunakan untuk menentukan model graf berbobot.

B. Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini adalah: perangkat keras (*hardware*) berupa komputer dengan prosesor Intel Core 2 CPU T5500 1,66 GHz, memori 2,49 GB RAM, hard disk 320 GB dan monitor 15,4 inchi. Perangkat lunak (*software*) berupa sistem operasi Microsoft Windows XP dan program Delphi 7.

C. Jenis Penelitian

Penelitian ini merupakan penelitian kualitatif dalam bidang teknik elektro dan teknologi informasi khususnya bidang rekayasa perangkat lunak dan algoritma komputer. Penelitian ini melakukan eksperimen dengan cara merancang dan mengembangkan suatu perangkat lunak yang akan diterapkan pada proses pencarian pohon merentang minimum (*minimum spanning tree*) suatu graf berbobot dengan menggunakan algoritma Prim.

D. Tahapan Penelitian

Penelitian ini dilakukan dengan melalui tahapan-tahapan sebagai berikut:

1. Melakukan pengamatan dan pengumpulan data panjang atau jarak untuk model dari graf berbobot.
2. Instalasi program yang dibutuhkan serta pengaturannya.
3. Melakukan persiapan data yang telah ada sehingga dapat digunakan oleh program aplikasi.
4. Merancang model graf sesuai dengan data yang diperoleh, kemudian dari graf tersebut diberi bobot masing-masing berupa jarak atau panjang dengan menggunakan program Delphi 7.
5. Dengan menggunakan algoritma Prim, ditentukan pohon merentang minimum dari model graf berbobot, kemudian dihitung dan disimulasikan oleh program Delphi 7 untuk mendapatkan jumlah total panjang minimum, urutan pohon merentang minimumnya, dan waktu komputasi.
6. Melakukan pengujian dan menarik kesimpulan dari hasil pengujian tersebut.

E. Tahapan Proses Aplikasi Algoritma Prim untuk Menentukan Pohon Merentang Minimum Suatu Graf Berbobot dengan Menggunakan Program Delphi 7.

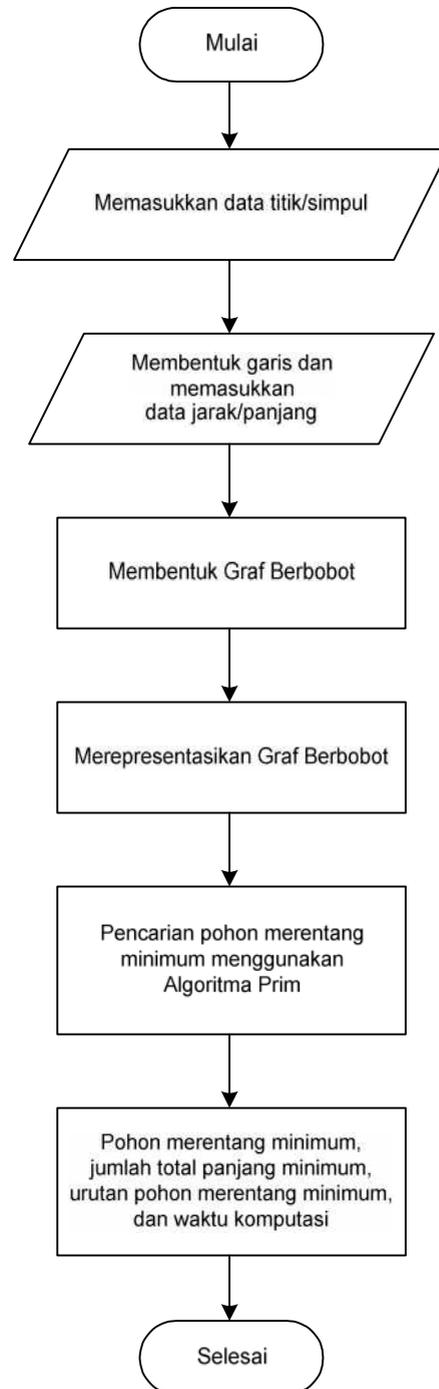
Data yang diperoleh mengenai jarak atau panjang yang menghubungkan titik-titik atau simpul-simpul dimasukkan ke dalam sistem untuk mendapatkan model graf berbobot. Titik-titik tersebut dihubungkan dengan garis sesuai dengan jarak masing-masing yang kemudian membentuk suatu graf berbobot dengan menggunakan program Delphi 7. Kemudian diproses dengan metode algoritma Prim untuk mendapatkan hasil berupa pohon merentang minimum, urutan pohon merentang minimum dan waktu komputasi dalam pencarian pohon merentang minimum tersebut. Keseluruhan tahapan proses algoritma Prim dalam mencari pohon merentang minimum dapat dijelaskan dengan gambar 5.

IV. Hasil dan Pembahasan

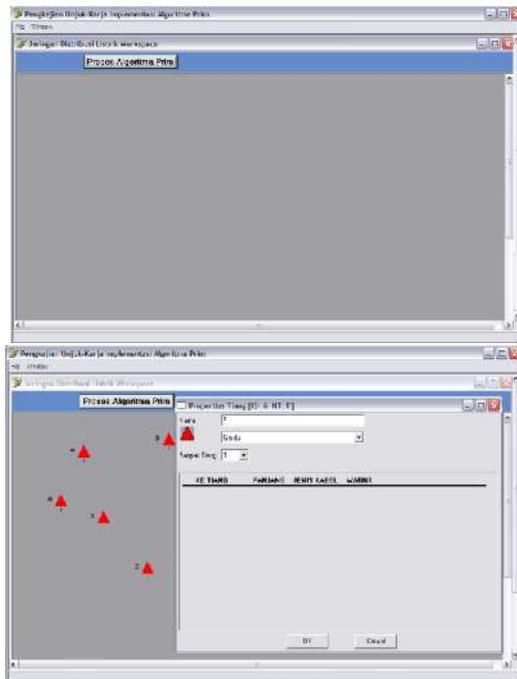
Dalam penelitian ini, program yang digunakan adalah program yang dibuat dengan bahasa pemrograman Delphi 7. Delphi adalah suatu program berbasis bahasa Pascal yang berjalan dalam lingkungan Windows. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut *Rapid Application Development* (RAD) yang telah membuat pemrograman menjadi lebih mudah (Madcoms, 2003). Delphi merupakan suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman *Object Oriented Programming* (OOP).

Implementasi sistem aplikasi algoritma Prim untuk menentukan pohon merentang minimum (*minimum spanning tree*) suatu graf berbobot dengan menggunakan program Delphi 7 melalui beberapa tahapan proses sampai didapatkan pohon merentang minimum suatu graf berbobot, informasi urutan pohon merentang minimumnya, informasi jumlah total panjang minimum, informasi waktu komputasi dalam mendapatkan pohon merentang minimumnya dan grafik hasil pengujian aplikasi algoritma Prim. Langkah-langkah

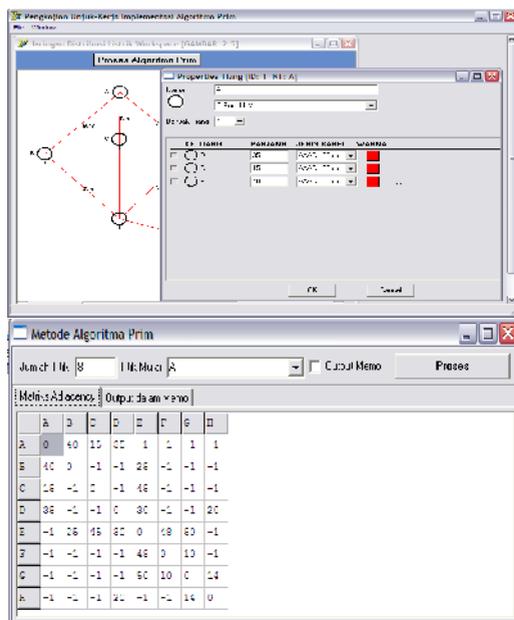
implementasi sistem ini dapat dilihat pada gambar 6, 7, dan 8.



Gambar 5. Flowchart tahapan pencarian pohon merentang minimum dengan menggunakan algoritma Prim



Gambar 6. Tampilan halaman awal sistem dan tampilan proses peletakan titik/simpul

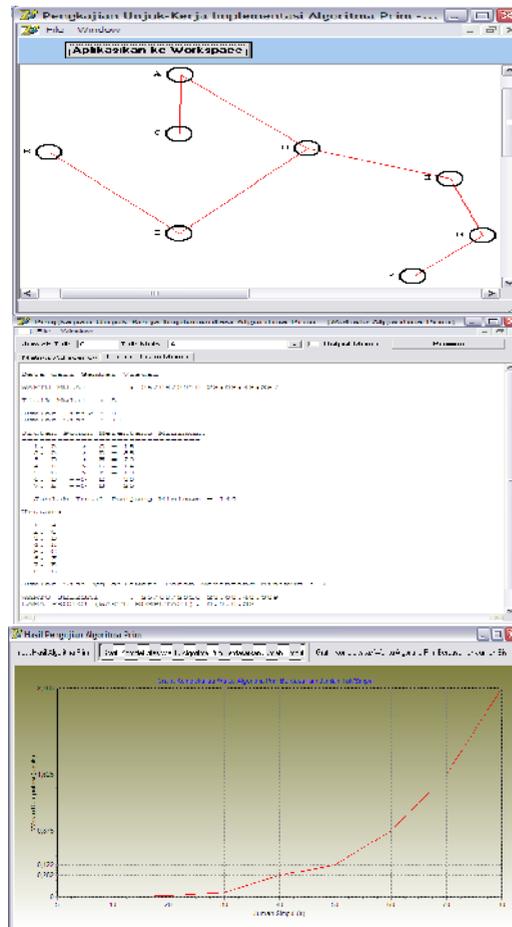


Gambar 7. Tampilan proses menghubungkan titik/simpul dengan garis dan tampilan pemilihan titik/simpul mulai pencarian pohon merentang minimum

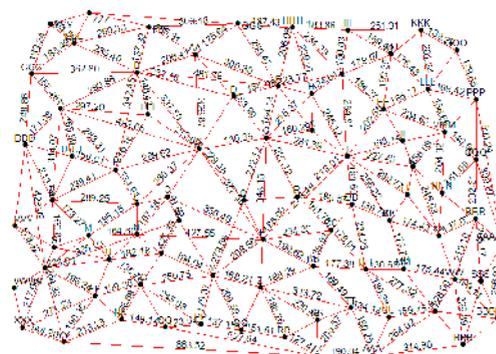
A. Pengujian Sistem

Contoh pengujian berikut adalah mencoba membuktikan aplikasi algoritma Prim dalam menentukan pohon merentang minimum (*minimum spanning tree*) dari model graf berbobot dengan 80 buah

titik/simpul dan 228 buah sisi. Model graf yang akan diuji pada pengujian ini dapat dilihat pada gambar 9.



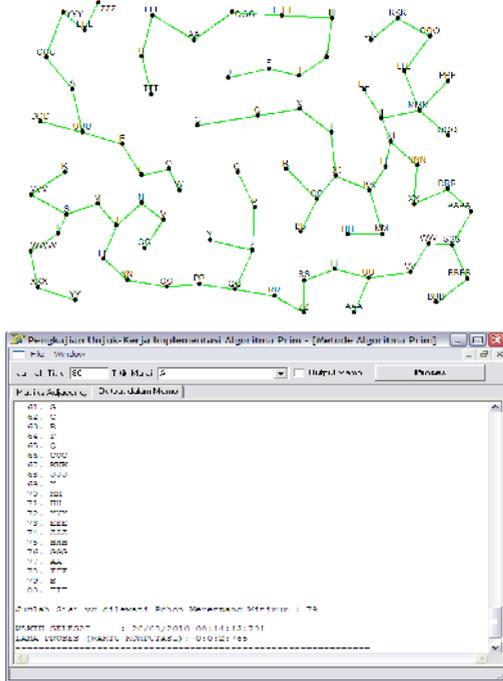
Gambar 8. Tampilan hasil proses pencarian pohon merentang minimum dengan algoritma Prim dan tampilan grafik hasil pengujian aplikasi algoritma Prim



Gambar 9. Model graf berbobot dengan 80 buah titik/simpul dan 228 buah sisi

Pohon merentang minimum yang terbentuk dari hasil proses algoritma Prim

terhadap model graf berbobot di atas dapat dilihat pada gambar 10. Hasil pengujian tersebut memperlihatkan urutan pohon merentang minimum dengan jumlah total panjang minimum sebesar 10958,1 dan waktu komputasi sebesar 2,765 detik.



Gambar 10. Pohon merentang minimum yang terbentuk dan hasil pengujiannya

Pengujian-pengujian dilakukan dengan model graf berbobot yang memiliki variasi dalam jumlah titik/simpul dan jumlah sisi. Pengujian-pengujian pada penelitian ini dilakukan agar mendapatkan waktu komputasi yang akurat untuk mengetahui unjuk-kerja aplikasi algoritma Prim. Dari hasil pengujian, maka dapat dibuatkan tabel aplikasi algoritma Prim seperti yang terlihat pada tabel 2.

B. Analisa Kemampuan Sistem

Program yang dibuat dengan bahasa pemrograman Delphi 7 digunakan untuk aplikasi algoritma Prim dalam menentukan pohon merentang minimum suatu graf berbobot. Program ini akan memodelkan graf, kemudian selanjutnya akan mencari pohon merentang minimumnya. Hasil yang diperoleh adalah pohon merentang minimum dari graf yang disimulasikan, urutan dan jumlah total panjang pohon merentang minimumnya serta waktu komputasi dalam

pencarian pohon merentang minimum tersebut.

Tabel 2. Hasil pengujian aplikasi algoritma Prim

JumlahTitik/ Simpul (n)	Jumlah Sisi (m)	Waktu Komputasi (detik)	Jumlah Total Panjang Minimum
5	8	0	1076,97
10	20	0	1821,2
20	49	0,015	3125,86
30	80	0,062	4221,54
40	104	0,282	5472,71
50	130	0,422	6826,18
60	163	0,875	8373,01
70	192	1,625	9790,13
80	228	2,765	10958,1

Kemampuan yang bisa dilakukan oleh sistem aplikasi algoritma Prim adalah sebagai berikut:

- Menampilkan graf berbobot dengan jarak/panjang sebagai bobotnya yang menghubungkan antara titik-titik/simpul-simpul.
- Menampilkan pohon merentang minimum secara visual dengan menggunakan algoritma Prim.
- Mampu menyimpan secara visual model graf berbobot dan hasil pohon merentang minimumnya.
- Menampilkan matriks ketetanggaan (*adjacency matrix*) yang berisi jarak/panjang yang menghubungkan antara titik-titik/simpul-simpul.
- Menampilkan informasi urutan pohon merentang minimum, jumlah total panjang minimum dan waktu komputasi dalam pencarian pohon merentang minimum.
- Menampilkan grafik hasil pengujian waktu komputasi dari aplikasi algoritma Prim.

Hasil analisa implementasi sistem aplikasi algoritma Prim dengan menggunakan program Delphi 7 dapat dijelaskan sebagai berikut:

1. Menggunakan 10 unit modul untuk implementasi algoritma Prim, yaitu unit

Global, unit Map, unit MDI_Map, unit MethodePrim, unit Properties, unit OutputVisual, unit Save, unit Open, unit Delete, dan unit HasilPrim. Masing-masing unit tersebut terdiri dari beberapa *procedure* dan *function* yang digunakan untuk mengolah data masukan dan menghasilkan keluaran berupa hasil proses algoritma Prim.

2. Menggunakan 9 model form yaitu form Map, form MDI_Map, form MethodePrim, form Properties, form OutputVisual, form Save, form Delete, form Open, dan form HasilPrim.
3. Proses algoritma Prim dikerjakan dengan struktur data *array* dan *record* yang digunakan untuk menyimpan titik-titik yang belum terhubung, menyimpan titik-titik yang sudah terhubung, menyimpan status dari titik-titik yang sudah terhubung di pohon merentang minimum, menyimpan jarak yang menghubungkan titik-titik, dan menyimpan hasil algoritma Prim dalam mencari pohon merentang minimum. Hasil akhir proses algoritma Prim akan disimpan ke dalam tabel-tabel yang ada dalam basis data yang digunakan dalam sistem ini.

V. Kesimpulan dan Saran

A. Kesimpulan

Setelah dilakukan serangkaian pengujian dan analisa dalam penelitian ini, maka dapat diambil kesimpulan sebagai berikut:

1. Program yang dibuat terbukti mampu mengaplikasikan algoritma Prim dalam menentukan pohon merentang minimum. Aplikasinya dilakukan dengan cara membentuk graf berbobot. Kemudian dicari pohon merentang minimumnya dengan menggunakan algoritma Prim dan dihasilkan informasi jumlah total jarak/panjang minimum yang menghubungkan semua titik/simpul dan waktu komputasi dalam pencarian pohon merentang minimum tersebut.
2. Hasil pengujian membuktikan bahwa program Delphi 7 dalam mengaplikasikan algoritma Prim untuk

suatu graf berbobot dengan jumlah titik/simpul (n) dari 5 sampai dengan 80 dan jumlah sisi (m) dari 8 sampai dengan 228, menghasilkan waktu komputasi yang cukup cepat yaitu hanya berkisar antara 0 sampai dengan 2,765 detik.

3. Berdasarkan hasil pengujian, waktu komputasi algoritma Prim dalam mencari pohon merentang minimum suatu graf berbobot akan bertambah naik seiring dengan bertambahnya jumlah titik/simpul dan jumlah sisi graf berbobot tersebut. Hasil pengujian aplikasi algoritma Prim bersifat kuadratik. Dengan waktu komputasi algoritma Prim tersebut yang bersifat kuadratik, maka dapat dibuktikan juga bahwa algoritma Prim termasuk dalam kategori algoritma yang baik atau algoritma yang efisien untuk memecahkan masalah pencarian pohon merentang minimum suatu graf berbobot.

B. Saran

Saran yang dapat diberikan adalah sebagai berikut:

- a. Diharapkan adanya penelitian lainnya yang menentukan kompleksitas waktu aplikasi algoritma Prim untuk graf berbobot.
- b. Perlunya dilakukan perbandingan antara beberapa algoritma lainnya untuk menentukan *minimum spanning tree* suatu graf berbobot agar diketahui algoritma mana yang lebih efisien.
- c. Perlunya membandingkan beberapa komputer dengan teknologi yang berbeda dalam menentukan *minimum spanning tree* dengan menggunakan algoritma Prim.

DAFTAR PUSTAKA

- Gloor, P. A., Johnson, D. B., Makedon, F., Metaxas, P., (1993), *A Visualization System for Correctness Proofs of Graph Algorithms*, http://www.wellesley.edu/CS/pmetaxas/visual_proofs.pdf, Computer Science Education, [diakses 19 Maret 2010].

- Greenberg, H. J., (1998), *Greedy Algorithm for Minimum Spanning Tree*, <http://glossary.computing.society.informs.org/notes/spanningtree.pdf>, University of Colorado, Denver, [diakses 19 Maret 2010].
- Kristanto, A., (2008), *Perancangan Sistem Informasi dan Aplikasinya*, Gava Media, Yogyakarta.
- Madcoms, (2003), *Pemrograman Borland Delphi 7*, Andi, Yogyakarta.
- Malik, J. J., (2006), *Kumpulan Latihan Pemrograman Delphi*, Andi, Yogyakarta
- Mehta, D. P., Sahni, S., (2005), *Handbook of Data Structures and Applications*, Chapman & Hall/CRC Computer and Information Science Series, United States of America.
- Munir, R., (2009), *Matematika Diskrit*, Edisi 3, Informatika, Bandung.
- Oetomo, B. S., (2002), *Perencanaan dan Pembangunan Sistem Informasi*, Andi, Yogyakarta.
- Pop, P. C., Zelina, I., (2004), *Heuristic Algorithms for the Generalized Minimum Spanning Tree Problem*, http://emis.library.cornell.edu/journals/AUA/acta8/Pop_Zelina.pdf, Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics (ICTAMI), Thessaloniki, Greece, [diakses 19 Maret 2010].
- Purbasari, I. Y., (2007), *Desain Dan Analisis Algoritma*, Edisi 1, Graha Ilmu, Yogyakarta.
- Purwanto, E. B., (2008), *Perancangan Dan Analisis Algoritma*, Edisi 1, Graha Ilmu, Yogyakarta.
- Zakaria, T. M., Prijono, A., (2006), *Konsep Dan Implementasi Struktur Data*, Informatika, Bandung.