ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

IMPLEMENTASI ALGORITMA PRIM DENGAN TEORI GRAPH PADA APLIKASI WPF GRAPH

***Trinanda Syahputra[#1], Muhammad Dahria[#2] , Iskandar Zulkarnain[#3]**
[#1]Program Studi Sistem Informasi, STMIK Triguna Dharma
[#2,3]Program Studi Sistem Komputer STMIK Triguna Dharma
E-Mail: [#1]trinandasyahputra@gmail.com

Abstrak

Teori graph merupakan konsep yang sudah cukup lama dipakai dan diterapkan pada banyak bidang. Makalah ini menyajikan bagaimana tataran konseptual graph, yaitu tentang gambaran umum, definisi graph, hingga sampai pada tataran implementasi, yaitu bagaimana konsep tersebut diterapkan dalam bidang ilmu komputer khususnya dalam Struktur Data dan menentukan minimum spanning tree (MST) yang banyak diaplikasikan dalam masalah TSP (Traveling Salesman Problem).

Algoritma Prim adalah sebuah algortima dalam teori graf yang mencari sebuah minimu spaning tree untuk menyelesaikan masalah-masalah TSP contohnya adalah penggantian sistem jaringan telepon atau rute jalur transportasi pengambilan surat ari kotak pos dan sebagainya.

Kata Kunci : Algoritma Prim, Graph, Struktur Data,TSP

Abstract

Graph theory is a concept that already long enough and applied TAKEN IN Many Fields . Papers singer presents how the conceptual level chart , ie ON Overview Sales manager , graphics Clearly, Up Up ON level of implementation , ie how the concept is implemented hearts Field of Computer Science in particular hearts Data structures and determine the minimum spanning tree (MST) The Many applied hearts TSP problem (Traveling Salesman Problem). Prim algorithm is A graph theory algorithms hearts minimu spaning The Finding A tree for a review solve the problem - the problem of TSP example is the replacement of the telephone network system OR the Line Transport Taking these ari letter box of Post and so on

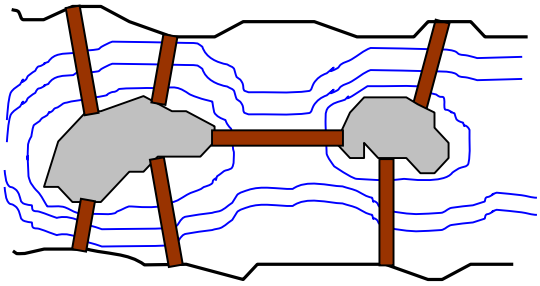
Keywords: *Prim Algorithm, Graph, Data Structures, TSP*

I. PENDAHULUAN

1. Latar Belakang

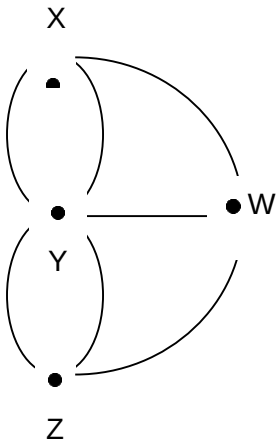
Algoritma Prim adalah sebuah algoritma dalam teori graf yang mencari sebuah *minimum spaning tree* untuk menyelesaikan masalah-masalah TSP contohnya adalah penggantian sistem jaringan telepon, rute pengambilan surat dari kotak pos dan sebagainya (Fadli, 2008: 13).

Teori graph diperkenalkan pada abad ke 18 oleh seorang matematikawan bernama Leonhard Euler. Euler mencoba memecahkan teka-teki yang dikenal dengan nama Masalah Jembatan Konigsberg. Terdapat tujuh buah jembatan yang menghubungkan dua pulau dan sebuah sungai, seperti yang ditunjukkan pada Gambar 1. Akan dicari sebuah lintasan yang melewati setiap jembatan tepat satu kali.



Gambar 1. Jembatan Konigsberg

Sebuah metode untuk mencari solusi dari masalah ini adalah dengan membentuk model dari jembatan Konigsberg yang dikenal sebagai *multigraph*, diperlihatkan pada Gambar 2. Sebuah *multigraph* memiliki dua elemen yaitu himpunan verteks (titik/node) dan himpunan *edge* (garis) yang menghubungkan antar verteks.



Gambar 2. Representasi *Multigraph* Jembatan Konigsberg

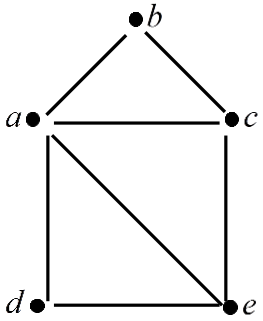
Titik-titik yang diberi label X, Y, Z, dan W pada Gambar 2 itulah yang disebut verteks, dan garis yang menghubungkan antar titik itulah yang disebut dengan *edge*. *Euler* menetapkan sebuah aturan yang bisa dipakai disemua *multigraph*, untuk mencari solusi dari masalah pada jembatan Konigsberg, aturan ini disebut dengan *Eulerian path*, yang berbunyi:

"Andaikan kita mempunyai sebuah multigraph sehingga untuk beberapa pasang verteks terdapat sebuah path (lintasan) diantara verteks-verteks tersebut. Multigraph tersebut memiliki Eulerian path jika dan hanya jika terdapat 0 atau 2 verteks yang mana banyak edge yang meninggalkan verteks tersebut berjumlah ganjil."

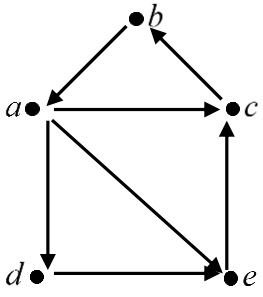
2. Tinjauan Pustaka

1. Teori Graph

Multigraph pada jembatan Konigsberg memiliki empat verteks, yang mana keempat verteks tersebut memiliki edge yang meninggalkan verteks tersebut berjumlah ganjil. Maka *multigraph* jembatan Konigsberg tidak memiliki Eulerian path. *Multigraph* yang ditunjukkan pada Gambar 3a tidak memiliki panah, sehingga disebut dengan undirected graph (graph tak berarah). Sebaliknya, *multigraph* yang memiliki panah disebut dengan directed graph (graph berarah)(Gambar 3b).



Gambar 3a. Graph Tak Berarah



Gambar 3b. Graph Berarah

Definisi 1**.** Sebuah simple graph (undirected graph) adalah pasangan dari $G = (V, E)$ dimana:

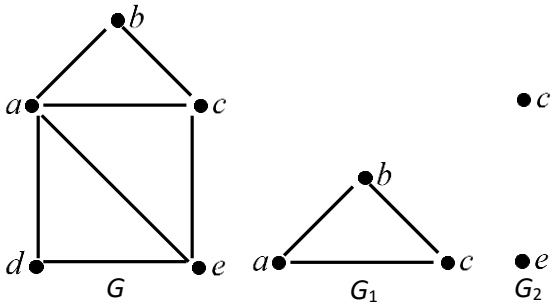
1. V adalah himpunan berhingga dari elemen yang disebut verteks
2. E adalah sebuah relasi yang irrefleksif dan simetri pada V.

Pasangan berurutan pada *E* disebut edge dari graph. Lebih spesifik, jika $e = (u, v) \in E$, dikatakan bahwa edge e adalah antara u dan v (dan juga antara v dan u), dan dikatakan bahwa u *adjacent* ke v. Lebih jauh, dapat dikatakan bahwa e *incident* ke u (dan juga v). Karena *E* simetri, maka kita dapat menotasikan e sebagai pasangan tak berurut $\{u, v\}$.

Andaikan $G = (V, E)$ sebuah graph. Dengan u, v verteks. *Degree* dari v, dinotasikan dengan $d(v)$, adalah jumlah *edge* yang *incident* ke v. Karena sebuah edge harus *incident* ke dua verteks, maka muncullah Teorema 1.

Teorema 1**.** Andaikan $G = (V, E)$

$$\sum_{v \in V} d(v) = 2|E|$$



Gambar 4. Graph G dan dua subgraphnya G_1 dan G_2

Sebuah subgraph dari graph $G = (V, E)$ adalah sebuah graph $G' = (V', E')$ sehingga $V' \acute{I} V$ dan $E' \acute{I} E$. Subgraph $G' = (V', E')$ disebut sebagai *spanning* subgraph jika $V = V'$. Gambar 4 menunjukkan dua subgraph, G_1 dan G_2 dari graph G.

2. Pohon *(Tree)*

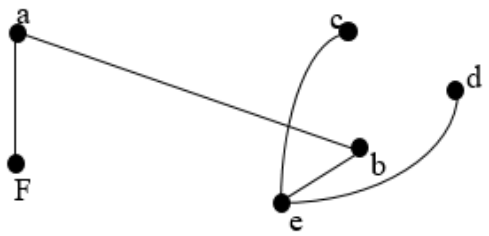
Pohon merupakan sebuah graf terhubung yang tidak mengandung sirkuit.Konsep pohon (*tree*) dalam teori graf merupakan konsep yang sangat penting, karena terapannya berbagai bidang ilmu. Oleh karenanya antara pohon *(tree)* sangat erat hubungannya dengan teori graf (Wahyudin, 2009).

Definisi pohon adalah graf tak berarah terhubung yang tidak mengandung sirkuit, menurut definisi tersebut, ada dua sifat penting pada pohon yaitu terhubung dan tidak mengadung sirkuit (Wahyudin, 2009).

Pohon *(tree)* merupakan graf dimana dua simpul memiliki paling banyak satu lintasan yang menghubungkannya. Pohon seringkali memiliki akar.Karena setiap simpul pada pohon hanya memiliki satu lintasan akses dari setiap simpul lainnya, maka tidak mungkin bagi sebuah lintasan untuk membentuk simpul *(loop)* atau siklus *(cycle)* yang secara berkesinambungan melalui serangkaian simpul (Wahyudin, 2009).

Pohon adalah graf tak-berarah terhubung dan tidak mengandung sirkuit.

Contoh pohon (Hernawati, 2008).



Sifat yang penting pada pohon adalah terhubung dan tidak mengandung sirkuit. Pohon dinotasikan sama dengan, *T= (V,E)*

Keterangan :

T : Tree

V: Vertices atau node atau vertex atau simpul, *V* merupakan himpinan tidak kosong.

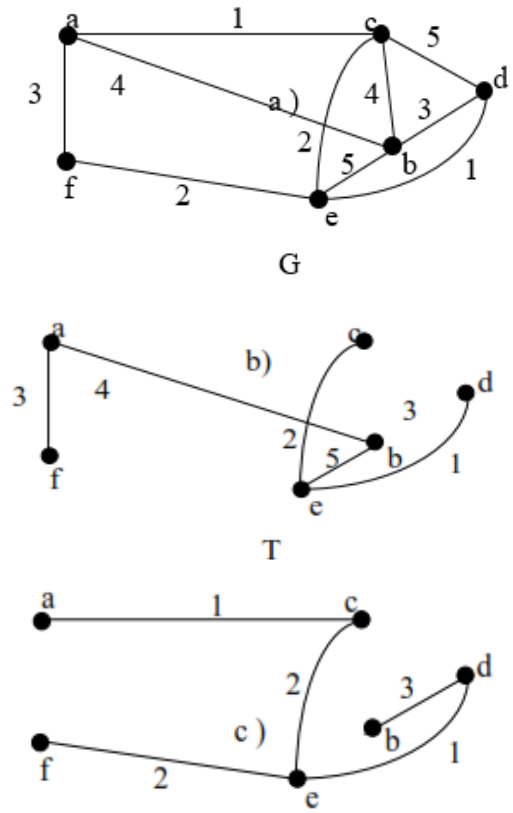
V= {v1,v2,…,vn}

E : *Edges* atau area atau sisi yang menghubungkan simpul

E= {e2, e2,…en}

3. Pohon Merentang *(Spanning Tree)*

Pohon rentang suatu graf *G* terhubung adalah subgraf *G* yang merupakan pohon dan memuat semua titik dalam *G.*Misalkan *G = (V,E)* adalah graf tak berarah terhubung yang bukan pohon, artinya di *G* terdapat sirkuit. *G* dapat diubah menjadi *T = (V,E)* dengan cara memutuskan salah satu sisi pada sirkuit-srikuit yang ada. Caranya yaitu dengan memutuskan salah satu sisi pada sirkuit hingga tidak ada sirkuit pada *G*. jika di *G* tidak lagi ada sirkuit maka pohon *T* ini disebut dengan pohon merentang. Disebut pohon merentang karena semua simpul pada pohon *T* sama dengan sinya pada graf *G* (Hernawati, 2008). Contoh pembentukan pohon merentang.



Gambar 5. Graf G, T1 dan T2 adalah pohon merentang dari Graf G

Keterangan :

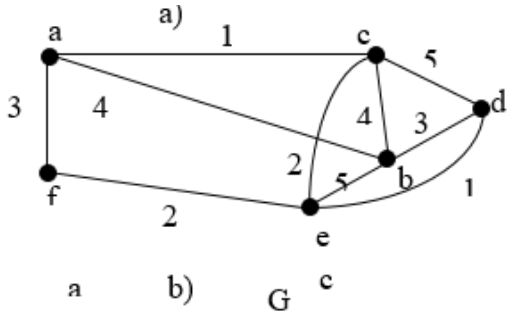
a. *T1* dan *T2* merupakan pohon merentang dari graf *G*.
b. Pohon merentang *T1* dibentuk dengan cara menghapus sisi *{(a,c),(b,c),(b,d),(c,d),(e,f)}* dari graf *G*.
c. Pohon merentang *T2* dibentuk dengan cara menghapus sisi *{(a,f),(a,b),(b,c), (b,e).(c,d)}* dari graf *G* (Hernawati, 2008).

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

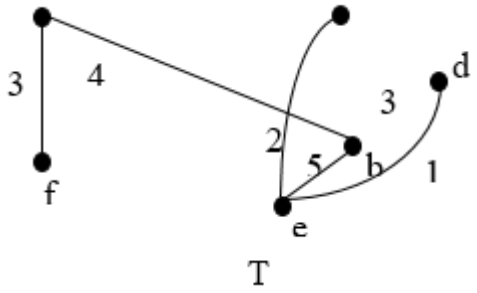
Algoritma pohon rentang minimum ditemukan dalam berbagai bidang aplikasi seperti, membutikan solusi optimal dari algoritma *greedy,* solusi perkiraan untuk masalah pohon rentang *minimum*, mendefinisikan kelompok dalam satu set data dan lainnya (Vikas, 2010).

Minimum spanning tree (disebut juga dengan MST) adalah mencari sebuah *spanning tree* dengan jumlah bobot *(weight)* minimal dari sebuah *graph* yang terhubung *(connected).* Masalah ini sama seperti pada *traveling salesman problem* (Wirdasari, 2011).

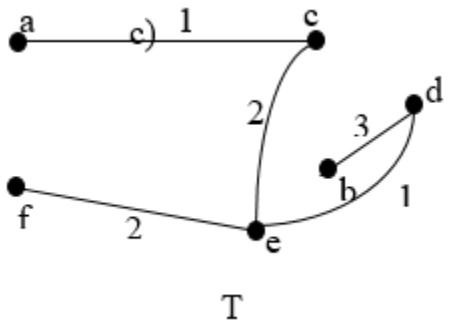
Jika G pada gambar 2.13 merupakan graf berbobot, maka bobot pohon merentang *T1* dan *T2* didefinisikan sebagai jumlah bobot semua sisi *T1* atau *T2*.Diantara pohon merentang yang ada pada *G*, yang paling penting adalah pohon merentang dengan bobot minimum atau *Minimum Spanning Tree* (MST). Contoh aplikasi MST yang sering digunakan adalah pemodelan proyek pembangunanjalan raya menggunakan graf. MST digunakan untuk memilih jalur dengan bobot terkecil yang akan diminimalkan biaya pembangunan jalan (Hernawati, 2008).
Contoh graf dan pohon berbobot :



Gambar 6. Graf G Berbobot dan T1 adalah Pohon Merentang Berbobot dari Graf G



Gambar 6. Graf G Berbobot dan T1 adalah Pohon Merentang Berbobot dari Graf G



Gambar 7. Graf T2 adalah Pohon Merentang Berbobot dari Graf G

Keterangan :

Dari graf berbobot *G,* kita harus menentukan pohon merentang mana yang paling minimum.Apakah *T1* atau *T2*. Hal tersebut yang akan dicari dengan membangun pohon merentang minimum (Hernawati, 2008).

5. Algoritma Prim

Algoritma Prim adalah algoritma dalam teori graf yang mencari pohon rentang minimum untuk sebuah graf berbobot yang terhubung. Ini berarti menemukan subset dari tepi yang membentuk sebuah pohon yang mencakup setiap titik, di mana berat total semua tepi di pohon diminimalkan. Jika grafik tidak terhubung, maka ia menemukan hutan rentang minimum (pohon rentang minimum untuk setiap komponen terhubung) (Subadra, 2011: 17).

Konsep dasar yang digunakan dalam algoritma Prim adalah pada setiap langkah, pilih sisi dari graf *G* yang berbobot minimum, tetapi sisi tersebut tidak membentuk sirkuit di *T* (Prima. P, 2010).

Langkah-langkah algoritma Prim :

a. Lakukan pengurutan terhadap setiap sisi di graf *G* mulai dari sisi dengan bobot terkecil.
b. Pilih sisi *(u,v)* yang mempunyai bobot minimum yang tidak terbentuk, yaitu ketika sirkuit di *T.* Tambahkan *(u,v)* ke dalam *T.*
c. Ulangi langkah 2 sampai pohon merentang minimum terbentuk, yaitu ketika sisi di dalam pohon merentang *T* berjumlah *n-1* (n adalah jumlah
d. simpulgraf *G*) penulisan algoritma Prim dalam bentuk notasi algoritmik *(pseudocode)*:

   Procudere Kruskal (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 rentang minimum T = (V,E)}

Deklarasi :*I, p, q, u, v* :*integer*

Algoritma :*{asumsi : sisi-sisi dari graf sudah diurut*

Menaik berdasarkan bobot}

T {} ←

While jumlah sisi di dalam T < n-1 do

Pilih sisi (u,v) dari E yang bobotnya Terkecil

If (u,v) tidak membentuk siklus di T then

T T … { (u,v) } ←

Endif
Endfor (Prima, 2010).

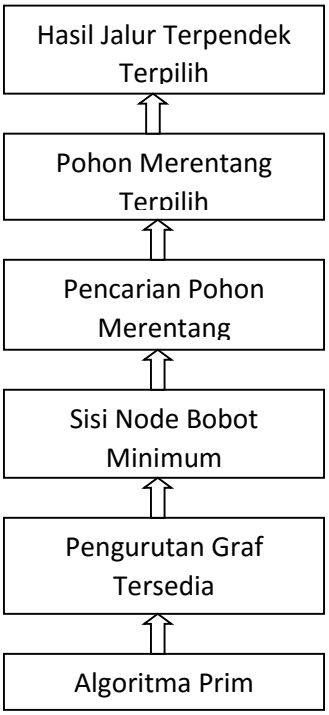
3. Tujuan dan Manfaat Penelitian

1. Adapun tujuan penelitian yang diperoleh dari penelitian ini adalah sebagai berikut:
   a. Memahami karakterikstik dari algoritma prim dengan teori grap pada aplikasi wpf Graph
   b. Memahami karakteristik teori graph pada aplikasi wpf Graph
2. Adapun manfaat penelitian yang diperoleh dari penelitian ini adalah sebagai berikut:
   a. Memperoleh pengetahuan algoritma prim dengan teori graph di matakuliah struktur data dan matematika diskrit
   b. Memperoleh pengetahuan teori graph dengan aplikasi graph pada matakuliah struktur data dan matematika diskrit

II. HASIL DAN PEMBAHASAN

1. Implementasi Algoritma Prim Dengan Teori Graph

Algoritma Prim adalah algoritma dalam teori graf yang mencari pohon rentang minimum untuk sebuah graf berbobot yang terhubung. Ini berarti menemukan subset dari tepi yang membentuk sebuah pohon yang mencakup setiap titik, dimana berat



Gambar 8. Arsitektur Algoritma Prim Pada Jalur Terpendek

Total semua tepi di pohon diminimalkan. Jika grafik tidak terhubung, maka ia menemukan hutan rentang minimum

Di dalam algoritma Prim terdapat beberapa teknik pengolahan data agar mendapatkan hasil yang bernilai. Beberapa langkah-langkah penyelesaian masalah sebagai berikut :

Algoritma Prim :

a. Lakukan pengurutan terhadap setiap sisi di graf mulai dari sisi dengan bobot terkecil.
b. Mempunyai sisi bobot *minimum* yang terhubung dengan *node.*
c. Pencarian pohon merentang minimum.
d. Adanya pohon merentang yang memiliki bobot *minimum*.

e. Lakukan pengurutan terhadap setiap sisi di graf mulai dari sisi dengan bobot terkecil.
f. Mempunyai sisi bobot minimum yang terhubung dengan *node.*
g. Pencarian pohon merentang minimum.
h. Adanya pohon merentang yang memiliki bobot minimum.

Dapat di lihat arsitektur algoritma Prim pada jalur terpendek seperti dibawah ini

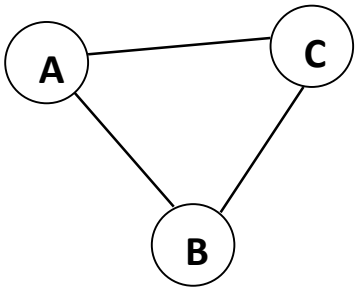
Dalam bidang ilmu komputer, sebuah graph dapat dinyatakan sebagai sebuah struktur data, atau secara spesifik dinamakan sebagai ADT (*abstract data type*) yang terdiri dari kumpulan simpul dan sisi yang membangun hubungan antar simpul. Konsep ADT graph ini merupakan turunan konsep graph dari bidang kajian matematika.

Pokok bahasan sebelumnya menjelaskan bahwa graph menampilkan visualisasi data dan hubungannya. Sedangkan jika berbicara masalah implementasi struktur data graph itu sendiri, isu utama yang dihadapi adalah bagaimana informasi itu di simpan dan dapat diakses dengan baik, ini yang dapat disebut dengan representasi *internal*.

Secara umum terdapat dua macam representasi dari struktur data graph yang dapat diimplementasi. Pertama, disebut *adjacency list*, dan diimplementasi dengan menampilkan masing-masing simpul sebagai sebuah struktur data yang mengandung senarai dari semua simpul yang saling berhubungan.

Yang kedua adalah representasi berupa *adjacency matrix* dimana baris dan kolom dari matriks (jika dalam konteks implementasi berupa senarai dua dimensi) tersebut merepresentasikan simpul awal dan simpul tujuan dan sebuah entri di dalam senarai yang menyatakan apakah terdapat sisi di antara kedua simpul tersebut.

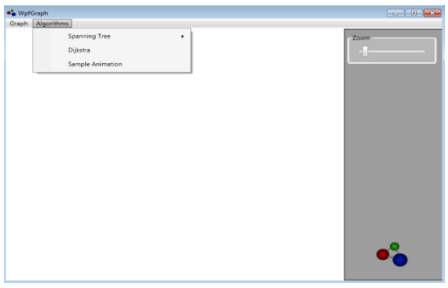
Dalam teori graph, *adjacency list* merupakan bentuk representasi dari seluruh sisi atau busur dalam suatu graph sebagai suatu senarai. Simpul-simpul yang dihubungkan sisi atau busur tersebut dinyatakan sebagai simpul yang saling terkait. Dalam implementasinya, *hash table* digunakan untuk menghubungkan sebuah simpul dengan senarai berisi simpul-simpul yang saling terkait tersebut.



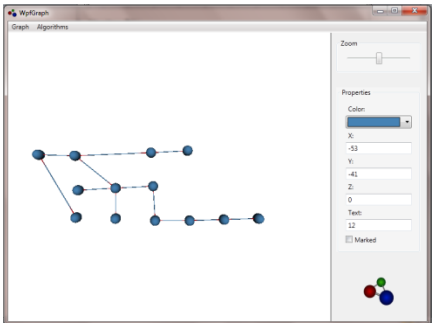
Gambar 9. Undirected Cyclic Graph

2. Algoritma Prim Pada Wpf Graph

Pada Wpf Graph terdapat pengolahan algoritma prim, yang mempermudah tahap penerapan algoritma, dapat dilihat pada gambar di bawah ini:



Gambar 10. *Interface* Menu *Algorithm* pada WpfGraph



Gambar 11. Hasil Algoritma Prim Menggunakan WpfGraph

III. KESIMPULAN DAN SARAN

1. Kesimpulan

a. Teori Graph merupakan salah satu cabang dari bidang Matematika Diskrit yang mempunyai banyak terapan di berbagai bidang.

b. Struktur data graph dan *Minimum Spanning Tree* merupakan bentuk implementasi dari teori graph yang mencakup definisi, dan hukum-hukum yang menyertainya.

c. Struktur data graph menggunakan representasi *internal* senarai ketetanggaan dengan alasan efisiensi penggunaan untuk komputasi, karena penggunaan matriks ketetanggan kurang efisisen dan cenderung boros untuk kasus jumlah sisi sedikit sedangkan matriks ketetanggaan yang dibentuk berupa matriks jarang (*sparse*)

Demikian hasil yang telah dimiliki dari implementasi algoritma prim dengan teori graph pada wpf graph.

2. Saran

a. Agar penelitian selanjutnya dapat mengembangkan algoritma lain sebagai solusi pemecahan masalah, sehingga dapat membandingkan algoritma tersebut terhadap algoritma yang penulis gunakan di dalam solusi pemecahan masalah. dapat menggunakan *tools* yang lebih akurat,

b. Agar terjadi sinkronisasi antara analisa data dan inplementasi sistem, perlu ada kajian yang lebih akurat terhadap jarak yang sebenarnya dengan jarak yang menjadi subjek penelitian.

DAFTAR PUSTAKA

[1] Baker, Roger. 2001. *Linear Algebra*. USA: Rinton Press.

[2] Bogart, Kenneth P., dan Stein, Cliff. 2002. *Discrete Math in Computer Science*. Dept. Of Computer Mathematics and Dept. Of Computer Science. Dartmouth College.

[3] Diestel, Reinhard. 2000. *Graph Theory*.New York: Springer-Verlag.

[4] Horn, R., dan Johnson, C. 1985. *Matrix Analysis*. Cambridge University Press.

[5] Kaw, Autar K. 2002. *Introduction to Matrix Algebra*. University of South Florida. http://www.eng.usf.edu/~kaw

[6] Munir, Rinaldi. 2003. *Diktat Kuliah IF2153. Matematika Diskrit Edisi Keempat.* Bandung: Penerbit ITB. http://www.informatika.org/~rinaldi

[7] Nobel, B., dan Daniel, J. 1977. *Applied Linear Algebra*. USA: Prentice Hall.

[8] Skvarcius, Romualdas., dan Robinson, William B. 1986. *Discrete Mathematics with Computer Science Applications*. California: The Benjamin/Cummings Publishing Company, Inc.