SIMULASI MINIMUM SPANNING TREE GRAF BERBOBOT MENGGUNAKAN ALGORITMA PRIM DAN ALGORITMA KRUSKAL

Mohammad Yasin*), Benny Afandi)**
email : my451n06@gmail.com
email : b2nafandi@gmail.com

ABSTRACT

There are so many problems that can be encountered by graphtechnique, especially in the realm of information technology (IT). One of the problems is to compute a*Minimum Spanning Tree (MST*). By using this kind of technique, it will be obtained a range of connected lines with its merits; e.g., the spanning tree of minimum cost, the least weight, and so on. There are commonly two algorithms to find minimum spanning tree (MST) namely Prim's algorithm and Kruskal's algorithm. This research aims to examine whether or not Prim's algorithm and Kruskal's algorithm can be used to calculate minimum spanning tree (MST) of a connected weighted graph, to know more about the approximate time to compute minimum spanning tree (MST) of a connected weighted graph through the implementation of Prim's algorithm and Kruskal's algorithm, and to find out which one of these two algorithms work best to calculate minimum spanning tree (MST) of a connected weighted graph. The development of software application in calculating MST using Prim's algorithm and Kruskal's algorithm is operated by Windows 7, the so-called Free Pascal Program, which includes the steps of observing and data gathering, the installation of the program, designing graph model that the weights are in accordance with the data gathered by Free Pascal Program, simulating the model of MST weighted graph through Prim's algorithm and Kruskal's algorithm by using Free Pascal Program to calculate the longest edge, the order of MST, computation time, and, lastly, to test and conclude the test result. The data simulation uses the data of PDAM pipeline "PerumnasPatrangJember". The result shows that MST program used perfectly proved that Prim's algorithm and Kruskal's algorithm can be implemented to calculate the minimum spanning tree (MST) of a connected weighted graph. The MST program usedcan also simulate and calculate the length of the MST weighted graph through the implementation of Prim's algorithm and Kruskal's algorithm.Based on the test result, the run time of Kruskal's algorithm is shorter and/or faster than the Prim's algorithm to calculate a various MST weighted graph.

Keywords: *Simulation, Minimum Spanning Tree (MST), Prim's algorithm, Kruskal's algorithm, Free Pascal*

ABSTRAK

Banyak permasalahan yang dapat dimodelkan dengan menggunakan graf, khususnya di bidang teknologi informasi. Salah satunya adalah masalah dalam pencarian pohon merentang minimum (*Minimum Spanning Tree/ MST*). 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. Terdapat dua buah algoritma dalam membangun MST, yaitu algoritma Prim dan algoritma Kruskal. Tujuan yang ingin dicapai dalam penelitian ini adalah : untuk membuktikan/menguji bahwa algoritma Prim dan Kruskal dapat digunakan untuk menentukan pohon merentang minimum suatu graf berbobot, untuk mengetahui unjuk-kerja aplikasi simulasi algoritma Prim dan algoritma Kruskal berupa waktu komputasi dalam menentukan pohon merentang minimum suatu graf berbobot, untuk mengetahui algoritma mana yang memiliki efektifitas pemodelan terbaik antara algoritma Prim dan algoritma Kruskal dalam menyelesaikan kasus pohon rentang minimum. Pengembangan aplikasi MST menggunakan algoritma Prim dan Kruskal menggunakan perangkat lunak (*software*) berupa sistem operasi Windows 7 dan program Free Pascal dengan tahapan melakukan pengamatan dan pengumpulan data, instalasi program, merancang

model graf dengan bobot sesuai data yang diperoleh menggunakan Free Pascal, mensimulasikan model graf berbobot dengan algoritma Prim dan algoritma Kruskal oleh program Free Pascal untuk menentukan jumlah total panjang minimum, urutan pohon merentang minimumnya, dan waktu komputasi, serta melakukan pengujian dan menarik kesimpulan dari hasil pengujian tersebut. Data simulasi menggunakan data jaringan Pipa PDAM Perumnas Patrang Jember diperoleh hasil bahawa Program MST yang dibuat terbukti dapat mengaplikasikan algoritma Prim dan algoritma Kruskal dalam menentukan pohon rentang minimum pada suatu graf berbobot; Program MST yang dibuat terbukti dapat mensimulasikan graf dan menghitung data bobot jarak/panjang graf awal dan graf pohon rentang minimum dengan menggunakan algoritma Prim dan algoritma Kruskal; Berdasarkan hasil pengujian, waktu komputasi algoritma Kruskal lebih pendek atau cepat jika dibandingkan dengan algoritma Prim untuk berbagai jenis graf berbobot yang dianalisis

Keyword *: Simulasi, Minimum Spanning Tree (MST), Algoritma Prim, Algoritma Kruskal, Free Pascal*

*) Dosen tetap Pendidikan Matematika FKIP Univ. Islam Jember
**) Dosen tetap Pendidikan Biologi FKIP Univ. Islam Jember

Pendahuluan

Latar Belakang Masalah

Banyak permasalahan yang dapat dimodelkan dengan menggunakan graf, salah satunya adalah masalah dalam pencarian pohon merentang minimum (*minimum spanning tree* / MST). 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 (Nugraha, 2011).

Nugraha (2011) telah mengembangkan sebuah aplikasi algoritma Prim untuk menentukan *minimum spanning tree* (MST) suatu graf berbobot dengan menggunakan pemrograman berorientasi objek. Hasil peneli tian tersebut hanya berfokus pada algoritma Prim dalam menyelesaikan masalah MST, sehingga perlu dikembangkan dan dibandingkan lagi dengan algoritma Kruskal yang juga mampu menyelesaikan masalah MST, dengan memban dingkan efektivitas waktu penyelesaian masalah.

Selanjutnya, akan dikembangkan sebuah aplikasi simulasi untuk melakukan perhitungan dan memonitoring terhadap suatu sistem yang

dapat membantu memecahkan permasalahan yang ada hubungannya dengan penentuan rute atau jalur dengan menggunakan pohon merentang minimum menggunakan algoritma Prim dan algoritma Kruskal.

Rumusan Masalah

Sesuai dengan ilustrasi yang diuraikan pada pendahuluan, maka rumusan masalah dalam penelitian ini adalah

1. Bagaimanakah algoritma Prim dan Kruskal diaplikasikan untuk menentukan pohon merentang minimum pada suatu graf berbobot?
2. Bagaimanakah program bantu yang dibuat dalam penelitian ini mampu mensimulasikan graf dan menghitung data jarak/panjang dengan menggunakan metode algoritma Prim dan Kruskal?
3. Algoritma manakah yang memiliki efektifitas pemodelan terbaik antara algoritma Prim dan algoritma Kruskal dalam menyelesaikan kasus pohon rentang minimum.

Tujuan Penelitian

Sesuai dengan latar belakang dan rumusan masalah di atas, tujuan yang ingin dicapai dalam penelitian ini adalah :

1. untuk membuktikan/menguji bahwa algoritma Prim dan Kruskal dapat digunakan untuk menentukan pohon merentang minimum suatu graf berbobot
2. untuk mengetahui unjuk-kerja aplikasi simulasi algoritma Prim dan algoritma Kruskal berupa waktu komputasi dalam menentukan pohon merentang minimum suatu graf berbobot.
3. Untuk mengetahui algoritma mana yang memiliki efektifitas pemodelan terbaik antara algoritma Prim dan algoritma Kruskal dalam menyelesaikan kasus pohon rentang minimum.

Tinjauan Pustaka

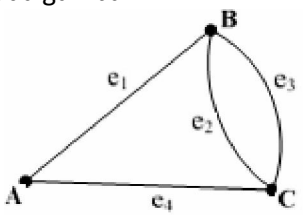
Algoritma

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 (Yulikuspartono, 2004). 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.

Graf

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, 2012). 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) …….. (1)

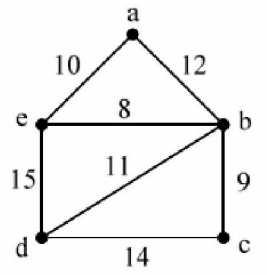
E = (e1, e2, e3, e4); bisa ditulis {(A,B),(B,C),(B,C),(A,C)} ……. (2)

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, 2012). 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*).

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

Matriks Ketetanggaan (*Adjacency Matrix*)

Matriks ketetanggan atau matriks kedekatan adalah representasi graf yang paling umum. Matriks ketetanggaan graf G adalah matriks yang berukuran n x 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 (Munir, 2012). 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 ketetanggan 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

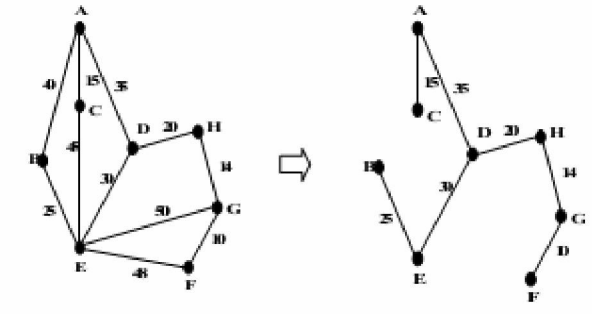
Tanda "∞" menyatakan bahwa tidak ada sisi dari simpul i ke simpul j sehingga a_{ij} dapat diberi nilai tak berhingga dan angka "0" menyatakan bahwa simpul saling berhubungan dengan simpulnya sendiri.

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, 2012).

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.

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 merentangnya adalah augmented dan dipilih simpul yang nilainya paling kecil dari semua simpul yang ada (Nugraha, 2011).

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.

Misalkan T adalah pohon merentang yang sisi-sisinya diambil dari graf G. Langkah-langkah dalam algoritma Prim adalah sebagai berikut:

1. Ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T.
2. Pilih sisi e yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi e tidak membentuk sirkuit di T. Masukkan e krdalam T.
3. Ulangi 2 sebanyak n-2 kali. (Munir, 2012)

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 ← {(p,q)}
for i ← 1 to n-1 do
Pilih sisi (u,v) dari E yang bobotnya terkecil namun bersisian dengan suatu simpul didalam T
T ← T ∪ {(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. (Nugraha, 2011)

Algoritma Kruskal

Pada algoritma Kruskal, sisi-sisi didalam graf diurut terlebih dahulun berdasarkan bobotnya dari kecil ke besar. Sisi yang dimasukkan kedalam himpunan T adalah sisi graf G sedemikian sehingga T adalah pohon. Pada keadaan awal, sisi-sisi sudah diurut berdasarkan bobot membentuk hutan (*forest*), masing-masing pohon dihutan hanya berupa satu buah simpul. Hutan tersebut dinamakan hutan merentang (*spanning forest*). Sisi dari graf G ditambahkan ke T jika ia tidak membentuk siklus di T (Munir, 2012).

Algoritma Kruskal adalah sebagai berikut:

1. Sisi-sisi graf diurut menaik berdasarkan bobotnya.
2. T masih kosong.
3. Pilih sisi e dengan bobot minimum yang tidak membentuk sirkuit di T. Masukkan e kedalam T.
4. Ulangi langkah 2 sebanyak n-1 kali.

Pseudo-code algoritma Kruskal adalah sebagai berikut :

Procedure 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 merentang minimum T = (V, E') }

Deklarasi
    i, p, q, u, v : integer

Algoritma
{Asumsi: sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya}
T ← {}
While jumlah sisi T < n-1 do
    e ← sisi (u,v) didalam E yang bobotnya terkecil
    E ← E – {e} {e sudah dipilih, jadi buang e dari E}
    If e tidak membentuk siklus di T then
        T ← T∪{(u,v)} masukkan e kedalam T yang sudah terbentuk}
    Endif
EndWhile

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. (Munir, 2012)

Metode Penelitian

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.

Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini adalah: perangkat keras (*hardware*) berupa komputer dengan prosesor Intel Core i3, memori 2 GB RAM, hard disk 320 GB, dan monitor 15,4 inchi. Perangkat lunak (*software*) berupa sistem operasi Microsoft Windows 7 dan program Free Pascal.

Jenis Penelitian

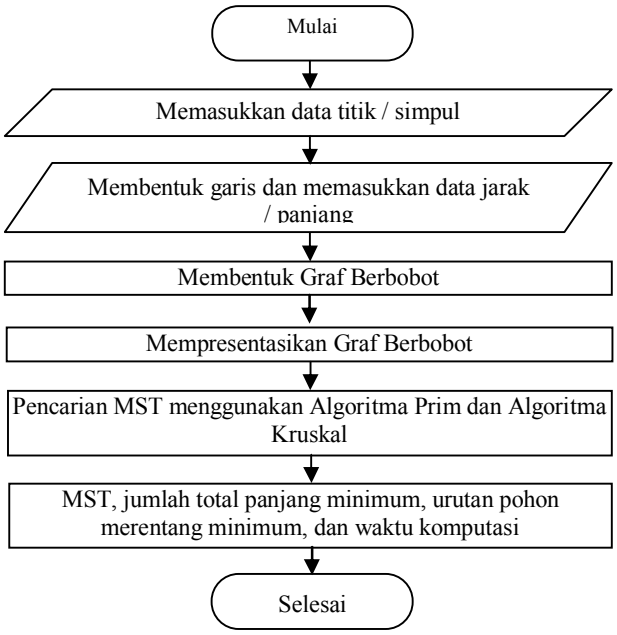
Penelitian ini merupakan penelitian kualitatif dalam bidang matematika komputasi khususnya bidang rekayasa perangkat lunak dan algoritma komputer. Penelitian ini melakukan eksperimen dengan cara merancang dan mengembangkan suatu perangkat lunak yang akan diterapkan pada proses pencarian pohon merentang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

Tahapan Penelitian

Penelitian ini dilakukan dengan melalui tahapan-tahapan sebagai berikut:

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

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 Free Pascal. Kemudian diproses dengan metode algoritma Prim dan algoritma Kruskal 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 dan algoritma Kruskal dalam mencari pohon merentang minimum dapat dijelaskan dengan gambar 4.



Gambar 4. Flowchart MST dengan Algoritma Prim dan Algoritma Kruskal

Hasil dan Pembahasan

Dari hasil penelitian dengan mengembangkan source code program Pascal untuk Algoritma Prim dan Algoritma Kruskal dengan menggunakan program Free Pascal adalah sebagai berikut.

Hasil Penelitian

Pembuatan Program MST

Pengembangan kode program Algoritma Prim dan Kruskal diawali dengan melakukan uji coba kode program dari berbagai sumber. Uji coba kode program menggunakan 2 contoh data yaitu :

Contoh 1:

```
4 5
1 2 10
1 3 14
1 4 15
2 3 12
3 4 9
```

Contoh 2:

```
7 11
1 4 5
3 5 5
4 6 6
2 4 9
1 2 7
2 5 7
2 3 8
4 5 15
5 6 8
5 7 9
6 7 11
```

Dari hasil ujicoba tersebut terdapat beberapa kelemahan, yaitu :

a. Terdapat error kode program untuk algoritma Kruskal, terutama pada bagian sorting data;
b. Program tidak flexibel untuk penggunaan berulang-ulang, terutama pada input data program;
c. Program sulit dikembangkan lebih lanjut, karena struktur program yang tidak terpisah-pisah antara bagian-bagian program, seperti input, proses, class, output.

Selanjutnya, kode program Algoritma Prim dan Kruskal dikembangkan lagi dengan memilah berdasarkan class program, class graf, class edge, dan class vertec. Output program Algoritma Prim dan Kruskal diuji dengan data diatas menghasilkan output program yang sesuai dengan perhitungan manual, sehingga kode program tersebut tinggal diuji coba dengan menggunakan data simulasi yang besar, atau dengan menggunakan data lapangan. Langkah selanjutnya, adalah mengembangkan validitas dan efektifitas output program, serta menyusun *user interface* dari program algoritma Prim dan Kruskal. Validitas output ditunjukkan dengan hasil output yang sama dengan hasil output perhitungan secara manual. Efektivitas algoritma Prim dan Kruskal ditunjukkan dengan kecepatan perhitungan dari masing-masing algoritma Prim dan Kruskal untuk data yang besar. *User interface* program dibuat untuk memudahkan pemakai dalam menggunakan program algoritma Prim dan Kruskal sebagai hasil akhir penelitian ini.

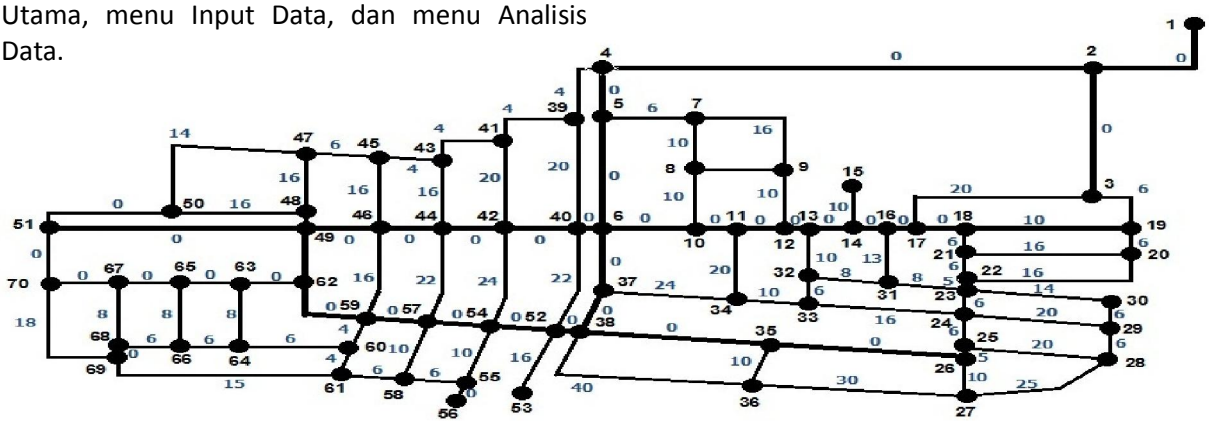
Ujicoba Program MST

Desain interface dari program MST dibuat untuk memudahkan pengguna dalam menggunakan program MST yang terdiri dari 3 bagian menu, yaitu bagian menu Halaman Utama, menu Input Data, dan menu Analisis Data.

Menu input data digunakan untuk membaca file data yang telah dibuat dengan ekstensi *.ini dengan model struktur data :

[VERTECS]
{ merupakan input data titik dengan format “kode_titik=nama_titik}
V1=A
V2=B
V3=C
...dst
[EDGES]
{merupakan bagian input data garis, data titik yg bersesuaian dengan ujung garis, serta bobot garis}
{format : kode_garis=bobot;kode_titik1;kode_titik 2}
E2=5;V3;V5
E1=5;V1;V4
E3=6;V4;V6
...dst

Ujicoba program akhir menggunakan data graf jaringan PDAM Perumnas Patrang Jember, dengan bobot panjang pipa PDAM. Graf jaringan pipa PDAM ini terdiri dari 70 vertec, 105 edge, dan dengan total bobot graf adalah 820, data ditampilkan sebagaimana pada gambar 5. Selanjutnya, graf jaringan pipa PDAM dianalisis dengan menggunakan program MST untuk menghasilkan graf pohon rentang minimum untuk menghasilkan jaringan pipa yang optimal dengan panjang pipa terpendek.



Gambar 5. Graf Jaringan Pipa PDAM Perumnas Patrang Jember

Hasil pengujian dengan menggunakan algoritma Prim dan algoritma Kruskal, sama – sama menghasilkan jaringan pipa minimum dengan 70 vertec, 69 edge, dan bobot akhir 239. Namun terdapat perbedaan hasil dari kedua algoritma, yaitu pada waktu komputasi, dimana waktu komputasi algoritma Prim = 6,41 detik dan algoritma Kruskal = 0,142 detik. Selanjutnya, peneliti melakukan simulasi menggunakan data random yang telah dibuat dengan berbagai jumlah vertec dan edge sebagaimana tabel berikut.

Tabel 2. Simulasi Program MST Algoritma Prim dan Algoritma Kruskal

No	Sebelum MST			Setelah MST					Ket
	V	E	Bobot	V	E	Bobot	T_{Prim}	T_{Krus}	
1	70	105	820	70	69	239	6,339	0,202	PDAM
2	100	195	21007	100	99	6592	34,294	0,452	Umum
3	100	99	10297	100	99	10297	17,607	0,044	Garis
4	30	435	45616	30	29	277	3,286	0,045	Lengkap

Pembahasan

Minimum Spanning Tree (MST) atau pohon rentang minimum, memiliki beberapa algoritma, diantaranya adalah algoritma Prim dan algoritma Kruskal. Kedua algoritma ini memiliki perbedaan pada proses awal dari algoritmanya. Dari tabel 2 diatas terlihat bahwa secara umum, algoritma Kruskal bisa berjalan lebih cepat dibanding algoritma Prim, baik untuk graf lengkap, graf sisi, maupun graf garis.

Sebenarnya, kecepatan program Kruskal dalam eksekusi MST terlihat dari tahapan masing – masing algoritma itu sendiri. Prim berorientasi pada pencarian simpul, sedangkan kruskal pada pencarian sisi dengan terlebih dahulu melakukan pengurutan bobot sisi. Jadi meskipun pada sebuah graf terdapat lebih banyak sisi dari pada simpul, Kruskal tetap lebih cepat jika dibandingkan Prim.

Hasil analisa implementasi sistem aplikasi Program MST dengan menggunakan program Free Pascal ini dapat dijelaskan sebagai berikut :

1. Menggunakan model pemrograman class dan proses algoritma dikerjakan dengan struktur data array dan record yang digunakan untuk menyimpan titik-titik yang belum terhubung, menyimpan titiktitik 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 dalam mencari pohon merentang minimum.
2. Input data adalah membaca file input dengan ekstensi *.ini, dan output pembacaan input maupun output hasil MST Prim dan hasil MST Kruskal adalah berupa file text dengan ekstensi *.txt, sehingga output program bisa digunakan untuk keperluan lainnya.
3. Menggunakan 3 model tampilan menu yaitu Menu Utama, Menu Graf, dan Menu MST.

Kesimpulan dan Saran

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

1. Program MST yang dibuat terbukti dapat mengaplikasikan algoritma Prim dan algoritma Kruskal dalam menentukan pohon rentang minimum pada suatu graf berbobot.
2. Program MST yang dibuat terbukti dapat mensimulasikan graf dan menghitung data bobot jarak/panjang graf awal dan graf pohon rentang minimum dengan menggunakan algoritma Prim dan algoritma Kruskal.

3. Berdasarkan hasil pengujian, waktu komputasi algoritma Kruskal lebih pendek atau cepat jika dibandingkan dengan algoritma Prim untuk berbagai jenis graf berbobot yang dianalisis.

Daftar Pustaka

Afrianto, I., Jamilah, E. W. 2012. *Penyelesian Masalah Minimum Spanning Tree menggunakan Ant Colony System* (ACS). Jurnal Ilmiah Komputa. Volume 1. Nomor 2. Bulan Oktober 2012. Bandung.

Hadiyanto. 2012. *Penentuan Pohon Rentang Minimum Pada Distribusi Jaringan Listrik Berdasarkan Kondisi Geografis Suatu Wilayah Dengan Algoritma Prim, Studi Kasus : Jaringan Listrik Distribusi Primer Kota Samarinda*. Jurnal Ilmiah Foristek. Volume 2. Nomor 1. Maret 2012.

Munir, R. 2013. *Matematika Diskrit*. Edisi kelima. Bandung: Informatika.

Nugraha, D.W. 2011. *Aplikasi Algoritma Prim Untuk Menentukan Minimum Spanning Tree Suatu Graf Berbobot dengan Menggunakan Pemrograman Berorientasi Objek*. Jurnal Ilmiah Foristek. Volume 1. Nomor 2. September 2011.

Wahid, Fathul. 2004. *Dasar-dasar Algoritma dan Pemrograman Pascal*. Edisi I. Yogyakarta : Andi.

Yulikuspartono. 2004. *Pengantar Logika dan Algoritma*. Edisi I. Yogyakarta : Andi.