

DIAGRAM ALIR ALGORITMA KRUSKAL DALAM MENENTUKAN MINIMUM SPANNING TREE

NinaZakiah

Dosen Sekolah Tinggi Agama Islam Negeri (STAIN)
Bengkalis, Riau, Indonesia

OK_Zakiah, Ms@yahoo.com

Abstract

The flow chart of the minimum spanning tree is represented in a weight matrix as a form of development of the link matrix. Determining the minimum spanning tree in this journal uses the Kruskal algorithm and is implemented in the form of a flow chart. This journal is a study of theories regarding the minimum spanning tree determination problem. With the problem approach, namely: graph theory, trees, determination of Minimum spanning tree, Kruskal's algorithm and flow diagrams. Furthermore, it is implemented in the flowchart of determining the Minimum spanning tree.

Keywords: *minimum spanning tree, kruskal's Algorithm, flow chart*

Abstrak

Diagram Alir minimum spanning tree direpresentasikan ke matriks bobot sebagai bentuk perkembangan dari matriks hubung. Menentukan Minimum spanning tree pada jurnal ini menggunakan algoritma kruskal dan diimplementasikan dalam bentuk diagram alir. Jurnal ini merupakan kajian dari teori-teori mengenai masalah Penentuan Minimum spanning tree. Dengan pendekatan masalah yaitu : teori graf, pohon, penentuan Minimum spanning tree, algoritma Kruskal serta diagram alir. Selanjutnya di implementasikan pada diagram alir penentuan Minimum spanning tree.

Keywords: minimum spanning tree, algoritma Kruskal, diagram alir

PENDAHULUAN

Ilmu pengetahuan serta teknologi yang berkembang pesat pada saat ini, tidak terlepas dari ilmu matematika sebagai ilmu yang menuntun penyelesaian masalah secara konseptual. Saat ini model matematika atau penalaran matematika banyak dimanfaatkan

sebagai alat bantu dalam penyelesaian masalah di berbagai bidang keilmuan.

Seorang matematikawan dari Swiss bernama Euler telah menerbitkan jurnal teori graf tahun 1736. permasalahan sehari-hari mengenai optimasi biasanya dapat diselesaikan menggunakan Minimum spanning tree, *Tree*(MST). seperti masalah mencari jarak pendek, biaya

termurah, dan tenaga seminimal mungkin dalam pembangunan.

Masyarakat saat memasang arus listrik di tiap ruangan dari rumahnya pasti menginginkan hasil yang efisien dengan biaya seminimal mungkin. Arus listrik dialirkan dari tiang listrik PLN sampai pada tiap rumah melalui kWh meter. Kabel listrik dipasang dari kWh meter guna mengalirkan listrik ke tempat-tempat lain.

Salah satu bentuk efisiensi biaya dalam pemasangan arus listrik ialah dengan penggunaan kabel listrik yang optimal. Salah satu cara optimalisasi penggunaan kabel listrik adalah dengan menggunakan konsep pohon retang minimum.

Dai dan Wu (2005) melakukan penelitian tentang algoritma Minimum spanning tree yaitu algoritma Prim, Kruskal dan biner, untuk menyelesaikan energy yang efisien masalah routing dalam jaringan

Cara lain menentukan Minimum spanning tree dapat menggunakan beberapa algoritma, seperti algoritma prim yang ditemukan oleh Robert C. Prim dan algoritma kruskal yang dirancang oleh Joseph Kruskal. Algoritma Prim berorientasi langsung pada pencarian sisi-sisi yang berbobot minimum berdasarkan simpul yang diambil, jadi tidak perlu mengurutkan sisi-sisi pada graf sesuai dengan bobotnya. Algoritma Kruskal menitik beratkan pemilihan sisi berdasarkan urutan bobotnya. Pengurutan sisi-sisi pada graf berdasarkan bobotnya sangat

diperlukan pada penyelesaian masalah yang menggunakan algoritma Kruskal. Sisi yang dipilih pada algoritma ini tidak boleh membentuk siklus namun tidak perlu bersisian pada simpul di T.

Pada prinsipnya algoritma prim tergolong lebih mudah dibandingkan algoritma kruskal. Tidak semua masalah dapat diselesaikan dengan optimal menggunakan algoritma prim. Pencarian Minimum spanning tree pada Graf yang mengandung simpul lebih banyak dari pada sisi atau graf yang mengandung beberapa pohon lebih efisien menggunakan Algoritma kruskal.

Algoritma kruskal dapat diterapkan secara manual dengan mengikuti tahap demi tahap sesuai prosedurnya. Pada graf yang besar (graf yang memiliki simpul dan sisi banyak) penerapan algoritma ini secara manual masih tergolong sulit dan memerlukan waktu yang lama. Kemampuan manusia untuk berkonsentrasi dalam penyelesaian algoritma kruskal tahap demi tahap tidaklah selalu baik, sehingga sering terjadi kesalahan.

Sejalan dengan perkembangan teknologi informasi, semakin bertambah pula kemampuan komputer dalam membantu penyelesaian permasalahan di berbagai bidang. Aplikasi pemrograman dapat dirancang sedemikian rupa untuk memudahkan penyelesaian masalah.

Tulisan ini membahas perancangan program yang dapat memudahkan penggunaan algoritma kruskal dalam menentukan Minimum

spanning tree dengan bantuan aplikasi Komputer.

Ada beberapa aplikasi pemrograman komputer yang dapat membantu penggunaan algoritma kruskal dalam menentukan minimum spanning tree dalam skala besar, seperti MATHLAB, VISUAL BASIC 6.0, pascal, dan beberapa aplikasi pemrograman lainnya.

Pemanfaatan aplikasi pemrograman dalam penentuan minimum spanning tree dilakukan dengan cara menerjemahkan langkah langkah algoritma ke dalam salah satu bahasa pemrograman yang dipilih. Untuk memudahkan menerjemahkan algoritma ke dalam bahasa pemrograman maka terlebih dahulu algoritma diterjemahkan kedalam bentuk diagram alir (Flow chart).

Penelitian ini dibatasi hanya sampai pada penerjemahan algoritma kruskal pada diagram alir berdasarkan latar belakang, pembahasan pada tulisan ini adalah diagram alir penentuan minimum Spanning tree dengan algoritma kruskal.

METODE

Penelitian ini adalah penelitian dasar atau teoritis. Metode yang digunakan adalah metode deskriptif, dengan menganalisis teori-teori yang relevan dengan permasalahan yang akan dibahas, serta berdasarkan pada studi kepustakaan..

HASIL DAN PEMBAHASAN

A. Penentuan Minimum spanning tree dengan algoritma kruskal

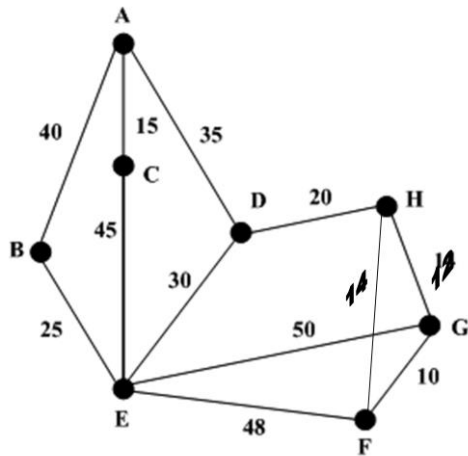
Penjabaran Algoritma kruskal dengan langkah langkah pengerjaan yang jelas seperti berikut

1. Merepresentasikan permasalahan yang ditemukan dalam bentuk graf
2. Setiap sisi pada graf diurutkan dari sisi dengan bobot terkecil sampai terbesar dan digambarkan ke bentuk tabel
3. Menggambar graf kosong yang meliputi semua simpul (vertex) pada langkah 1
4. Menggambar sisi dengan menarik garis lurus kedua simpul sehingga terbentuk sisi dengan bobot terkecil.
5. Hapus sisi yang telah di pakai dari table.
6. Ulangii langkah 4 dengan catatan graf tidak membentuk sikel.
7. Apabila graf berbentuk sikel maka Ulangi langkah 5.
8. Ulangi langkah 4 sampai graf memiliki jumlah sisi kurang satu dari jumlah simpul pada graf.

Contoh pencarian Minimum spanning tree pada suatu graf dengan algoritma kruskal secara manual.

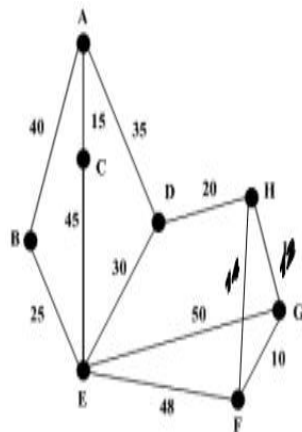
Contoh 1

Tentukanlah Minimum spanning tree dengan algoritma kruskal dari graf berbobot seperti gambar dibawah ini



Gambar 1 : Gambar graf berbobot contoh soal

1) Merepresentasikan permasalahan yang ditemukan dalam bentuk gambar.



Gambar 2. Contoh langkah 1

2) Membentuk tabel dengan daftar semua sisi pada graf diatas. Penulisan sisi dalam bentuk tabel di urutkan sesuai bobotnya yaitu dari yang terkecil sampai terbesar

Tabel 1 . tabel urutan sisi berdasarkan bobot

No	BOBOT	SISI
1	10	(F,G)
2	12	(G,H)
3	14	(H,F)
4	15	(A,C)
5	20	(D,H)
6	25	(B,E)
7	30	(D,E)
8	35	(A,D)
9	40	(A,B)
10	45	(C,E)
11	48	(E,F)
12	50	(E,G)

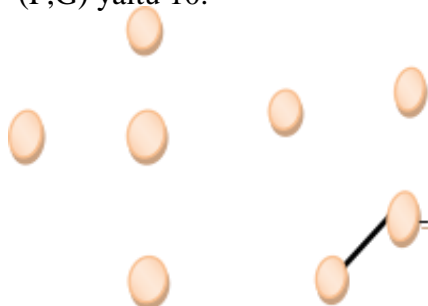
3) Membuat graf kosong meliputi semua vertex yang ada pada gambar



Gambar 3. Graf kosong (langkah 3)

4) Membuat sisi yang menghubungkan kedua vertex sehingga membentuk sisi dengan bobot terkecil. Sisi dengan bobot

paling rendah pertama adalah (F,G) yaitu 10.

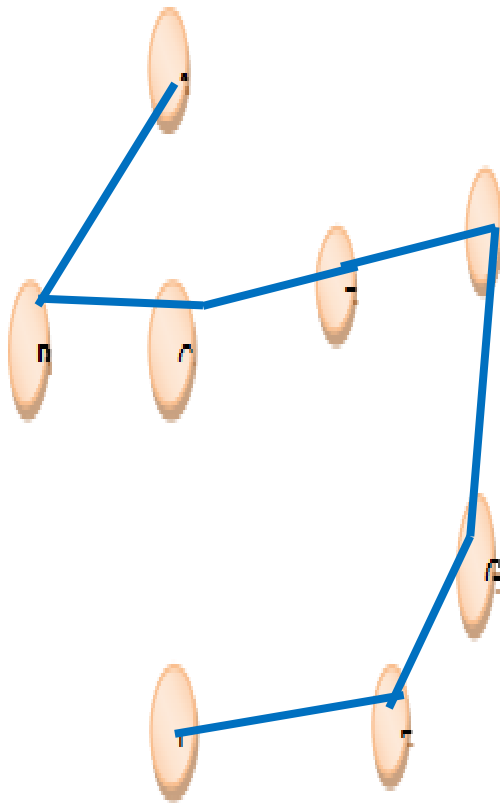


Gambar 4. Gambar graf langkah (4)

- 5) Menghapus sisi yang sudah dipakai, demikian sehingga sisi kedua terkecil merupakan sisi terkecil.
- 6) Mengulangi kembali dari langkah 4 dengan syarat graf tidak membentuk siklus. Sisi terkecil selanjutnya adalah (G,H) dengan bobot 12 jadi jumlah bobot graf tersebut $10+12=22$.
- 7) Melanjutkan langkah 5 yaitu Menghapus sisi dan bobot yang sudah dipakai dari daftar table, demikian sehingga sisi kedua terkecil merupakan sisi terkecil.
- 8) Mengulangi langkah 4 dengan syarat graf tidak membentuk siklus. Sisi terkecil selanjutnya adalah (H,F) dengan bobot 14, karena sisi (H,F) akan membentuk siklus jika digunakan maka sisi (H,F) diabaikan. Melanjutkan langkah lima yaitu Menghapus sisi dan bobot yang sudah dipakai dari daftar table, demikian sehingga sisi kedua terkecil merupakan sisi terkecil.
- 9) Mengulangi kembali dari langkah 4 dengan syarat graf tidak membentuk siklus. Sisi terkecil selanjutnya (A,C) dengan bobot 15 maka jumlah bobot graf yang terbentuk adalah $10+12+15=37$
- 10) Mengulangi langkah 5 yaitu Menghapus sisi dan bobot yang sudah dipakai dari daftar table. Demikian sehingga sisi kedua terkecil merupakan sisi terkecil. Mengulangi langkah 4 dengan syarat graf tidak membentuk siklus, sisi terkecil (D,H) dengan bobot 20 jadi jumlah bobot graf yang terbentuk $10+12+15+20=57$
- 11) Mengulangi langkah 5 yaitu Menghapus sisi dan bobot yang sudah dipakai dari daftar table. Demikian sehingga sisi kedua terkecil merupakan sisi terkecil.
- 12) Mengulangi langkah 4 dengan syarat graf tidak membentuk siklus, sisi terkecil (B,E) dengan bobot 25 jadi jumlah bobot graf yang terbentuk $10+12+15+20+25=82$
- 13) Mengulangi langkah 5 yaitu Menghapus sisi dan bobot yang sudah dipakai dari daftar table, demikian sehingga sisi kedua terkecil merupakan sisi terkecil.
- 14) Mengulangi langkah 4 dengan syarat graf tidak membentuk siklus. Sisi terkecil (D,E) dengan bobot 30 jadi jumlah bobot graf yang terbentuk $10+12+15+20+25+30=112$
- 15) Mengulangi langkah 5 yaitu Menghapus sisi dan bobot yang sudah dipakai dari daftar table,

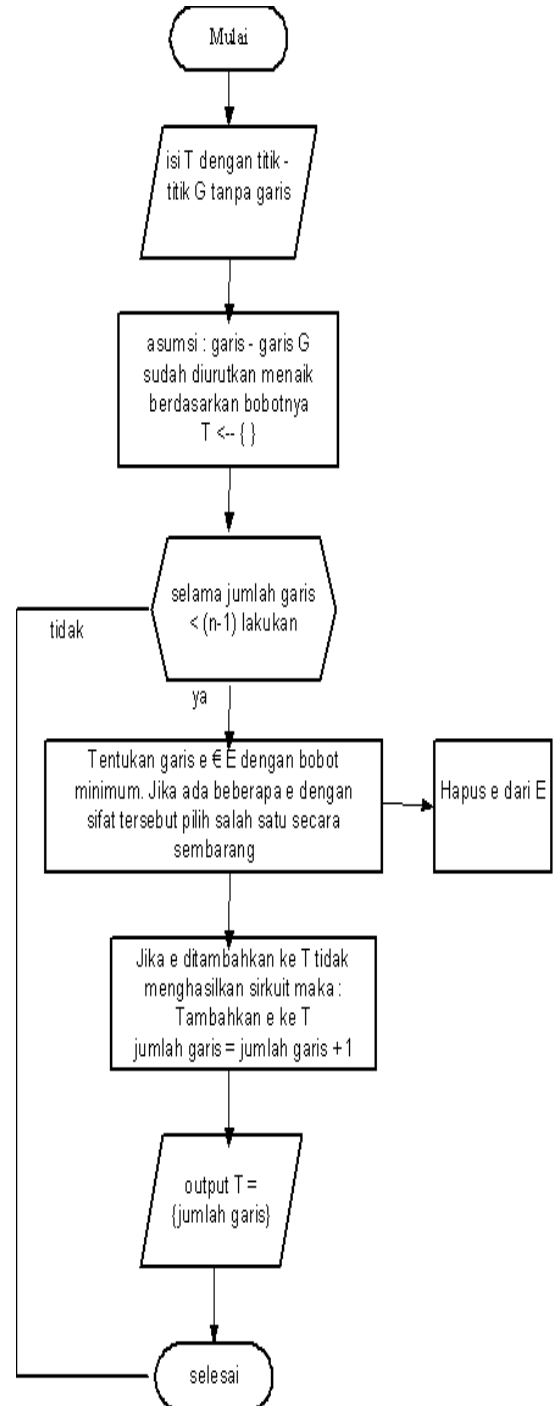
16) Mengulangi langkah 4 dengan syarat graf tidak membentuk siklus. Sisi terkecil (A,D) dengan bobot 35 jadi jumlah bobot graf yang terbentuk
 $10+12+15+20+25+30+35 = 147$

17) Langkah ini sudah selesai sini tidak di teruskan lagi karna jumlah sisinya sudah kurang satu dari simpulnya yaitu 7 buah dan jumlah bobot minimal dari pohon yang terbentuk adalah 147



Gambar 7. Graf akhir

B. Diagram Alir



SIMPULAN

Tujuan dari model penentuan lokasi server ini adalah peletakan server di lokasi yang efisien, sehingga dapat menghemat waktu perjalanan pelanggan menuju server terdekat dan waktu yang dihabiskan pelanggan pada

Server serta mempertimbangkan efisiensi biaya. Hasil dari model tersebut menyatakan bahwa peletakan efisien server pada node graph, dimana server p harus berada di beberapa node

DAFTAR RUJUKAN

- Bentley, Whitten. 2009. *System Analysis And Design Method*. Avenue of the Americas, NewYork.
- Chartrand, Gary & Lesniak, Linda. 1986. *Graphs & Digraphs*. 2nd edition, California : Greg Hubit Bookworks.
- Divisi Penelitian dan Pengembangan Madcoms, *Pemrograman Tingkat Lanjut dengan Visual Basic*, Andi Ofset, Yogyakarta 2005.
- Jek Siang Jong, *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer*, Andi, Yogyakarta, 2002
- Munir Rinaldi, (2003) *Matematika Diskrit*. Informatika. Bandung.
- Oetomo, B. S., (2002), *Perencanaan dan Pembangunan Sistem Informasi*, Andi, Yogyakarta.
- 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.
- Partono, Yulikus *Pengantar Logika dan Algoritma* , Andi Publisher , Yogyakarta,2008.
- Suarga, (2012), *Algoritma dan Pemrograman*, Andi Publisher, Yogyakarta.
- Sismoro Heri, *Pengantar Logika Informatika, Algoritma Dan Pemrograman*, Adi Publisher, Yogyakarta 2000
- Budayasa, I Ketut. 2007. *Teory Graph dan Aplikasinya*. Surabaya : University Press IKIP Surabaya.