

Pengujian Optimalisasi Jaringan Kabel *Fiber Optic* di Universitas Islam Indonesia Menggunakan *Minimum Spanning Tree*

Muchammad Abrori dan Najib Ubaidillah

Program Studi Matematika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga, Jl. Marsda Adisucipto No. 1 Yogyakarta, Indonesia

Korespondensi; Email: borymuch@yahoo.com

Abstrak

Universitas Islam Indonesia (UII) mengintegrasikan jaringan komputer kampus yang dibangun sejak 1995. Pengembangan jaringan komputer kampus UII terintegrasi menggunakan topologi star and fiber optic (FO). Mengingat topologi star adalah topologi yang membutuhkan banyak kabel, penelitian ini dilakukan untuk mengetahui dan mengkaji bagaimana penerapan grafik pada jaringan kabel FO kampus terpadu UII untuk meminimalisir biaya, karena jaringan kabel FO dapat dimodelkan. Dengan grafik dimana bangunan sebagai titik, sedangkan kabel FO yang terhubung ke masing-masing bangunan sebagai antrean. Jenis penelitian yang digunakan di sini adalah studi kasus, di mana pengumpulan data dilakukan dengan observasi, wawancara, dan dokumentasi. Penelitian ini menggunakan 4 algoritma, yaitu algoritma Kruskal, Prim, Boruvka dan algoritma Solin untuk menemukan Minimum Spanning Tree. Berdasarkan hasil penelitian yang telah dilakukan, kesimpulan tentang langkah pemecahan masalah optimalisasi algoritma jaringan kabel berbasis kampus FO kampus terintegrasi telah didapat. Dari keempat algoritma tersebut didapatkan hasil paling optimal panjang kabel FO sepanjang 4,700 meter dan dilalui kabel sepanjang 1.590 meter. Sedangkan hasil pengamatan yang dilakukan, diketahui bahwa jaringan komputer yang ada di kampus terpadu UII memiliki panjang kabel 6.120 meter dan panjang lintasan 2.050 meter. Hasil analisis menunjukkan bahwa hasil penelitian 23,2% lebih optimal daripada jaringan komputer yang ada di kampus terpadu UII.

Kata Kunci: Kabel serat optik; Pohon rentang minimum; Topologi Star; Jaringan komputer UII

Abstract

Universitas Islam Indonesia (UII) integrated campus computer network built since 1995. Development of UII integrated campus computer network is using a star topology and fiber optic (FO) cable. Considering that the star topology is the topology that requires a lot of wires, this study was conducted to determine and examine how the application of graph on the FO cable network UII integrated campus in order to minimize the cost, because FO cable network can be modeled by a graph where the buildings as points, while FO cable that connects to each building as a line. This type of research that is used here is a case study, in which data collection by observation, interviews, and documentation. This study used 4 algorithms, that is Kruskal algorithm, Prim, Boruvka and Solin algorithm to find the Minimum Spanning Tree. Based on the research that has been done, the conclusion about the troubleshooting steps of optimization UII integrated campus FO cable network based graph theory has been got. From the four algorithms obtained the most optimal results FO cable length 4.700 meters long and is 1.590 meters cable lines. While the results of observations made, it is known that the existing computer network in UII integrated campus has a cable length of 6.120 meters and 2.050 meters long track. The results of the analysis showed that the results of the study 23.2% more optimal than the existing computer networks in UII integrated campus.

Keywords: Fiber optic cable; Minimum spanning tree; Star topology; UII computer network

Pendahuluan

Teori graf merupakan salah satu pokok bahasan matematika diskrit yang telah lama dikenal dan banyak diaplikasikan pada berbagai bidang. Beberapa permasalahan sehari-hari yang dapat direpresentasikan

dalam graf, diantaranya adalah jaringan transportasi, jaringan komunikasi, jaringan komputer, dan dalam beberapa teori permainan.

Graf secara praktis terdiri dari 2 bagian yaitu himpunan titik yang sering disebut dengan himpunan titik/ simpul/ node ($V=Vertice$) dan himpunan garis atau himpunan sisi/ rusuk/ busur/ cabang ($E=Edge$) yang saling asing. Graf secara simbolik dapat dituliskan dengan bentuk $G(V, E)$. (Jong Jek Siang, 2002: 187).

Hubungan kabel-kabel (*cabling*) FO pada jaringan komputer UII dapat disajikan dengan menggunakan sebuah graf, dimana gedung-gedung di kampus UII digambarkan sebagai titik, sedangkan kabel FO sebagai garis yang menghubungkan tiap titik.

Penelitian ini bertujuan untuk mengetahui bagaimana penerapan graf pada jaringan kabel FO yang ada di Universitas Islam Indonesia dan mengetahui sejauh mana efisiensi jaringan kabel FO yang ada di UII agar berbiaya minimum.

Jaringan Komputer

Pengertian jaringan komputer

Jaringan komputer merupakan sistem yang terdiri dari gabungan beberapa perangkat komputer yang didesain untuk dapat berbagi sumber daya, berkomunikasi dan akses informasi dari berbagai tempat, antar komputer yang satu dengan komputer yang lain. (Madcoms, 2013: 1).

Macam-macam jaringan komputer

Jaringan komputer berdasarkan luas area terbagi menjadi 3 jenis yaitu *Lokal Area Network* (LAN), *Metropolitan Area Network* (MAN), dan *Wide Area Network* (WAN). (Madcoms, 2013: 5).

Jenis-jenis jaringan komputer berdasarkan topologi

Topologi merupakan istilah yang digunakan untuk menyatakan tentang cara bagaimana komputer terhubung dalam suatu jaringan. Topologi dibagi menjadi dua yaitu topologi fisik (topologi bus, topologi ring, topologi star, dan topologi mesh), dan topologi logic (*broadcast topology* dan *token passing topology*). (Wahana Komputer, 2005: 6).

Perangkat keras (hardware) jaringan komputer

Beberapa media *hardware* yang penting di dalam membangun suatu jaringan yaitu kabel, hub switch, repeater, bridge, router, ataupun modem, dll (Melwin Syafrizal, 2005: 21).

Teori Graf

Definisi graf

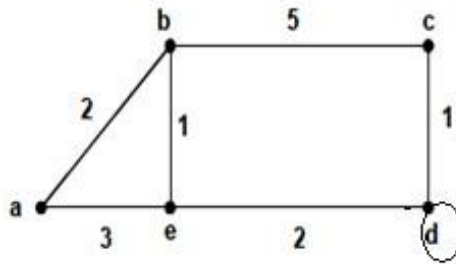
Graf G adalah pasangan $(V(G), E(G))$ dengan $V(G)$ adalah himpunan tidak kosong dan berhingga dari objek-objek yang disebut titik, dan $E(G)$ adalah himpunan (boleh kosong) pasangan tidak berurutan dari titik-titik berbeda dari $V(G)$ yang disebut sisi. (Abdussakir, 2009: 4) Dua buah titik pada graf tak-berarah G dikatakan bertetangga (*Adjacent*) jika ada sisi yang menghubungkan keduanya. Untuk sebarang sisi $e = (v_i, v_j)$ disebut bersisian (*incident*) dengan titik v_i dan titik v_j . Sebuah sisi yang menghubungkan sebuah titik ke dirinya sendiri disebut *loop*. Banyaknya sisi yang incident pada titik tersebut disebut derajat titik. (Kenneth H. Rosen, 1999: 445).

Graf terhubung

Graf tak-berarah G dikatakan terhubung jika ada dua titik v dan w dalam G terdapat *walk* (lintasan) dari v ke w . (Kenneth H. Rosen, 1999: 468)

Graf berbobot

Graf G disebut Graf berbobot jika setiap sisi e dari G diberi suatu bilangan non-negatif $w(e)$ yang disebut bobot atau panjang dari v . (Seymour Lipschutz, 2008: 141) Contoh 1.



Gambar 1 Graf terhubung, graf berbobot, loop di titik d, (b,c) adjacent, 5 incident (b,c).

Subgraf

Misalkan G graf. Graf H dikatakan subgraf dari G jika setiap titik di H adalah titik di G dan setiap sisi di H adalah sisi di G . Dengan kata lain, graf H adalah subgraf dari graf G jika $V(H) \subseteq V(G)$ dan $E(H) \subseteq E(G)$. (Abdussakir, 2009: 39)

Lintasan

Jalur dari suatu titik awal v_0 ke titik tujuan v_T di dalam suatu graf G merupakan barisan sebuah sisi atau lebih $(x_0, x_1), (x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n)$ pada G , dimana $x_0 = v_0$ dan $x_n = v_T$. Jalur yang digunakan tidak melakukan pengulangan titik maka jalur ini dinamakan lintasan (*path*). (Jong Jek Siang, 2002: 241)

Pohon

Pohon adalah sebuah graf yang mempunyai n buah titik, $n - 1$ sisi dan tidak mempunyai lintasan (*cycle free*) serta merupakan graf terhubung. Suatu pohon titik yang berderajat 1 dinamakan daun (*leaf*) atau titik terminal (*terminal vertex*), sedangkan titik yang berderajat lebih dari 1 disebut titik cabang (*branch vertex*) atau titik internal (*internal vertex*). (Samuel Wibisono, 2008: 159).

Hutan

Suatu hutan G adalah suatu graf tanpa siklus, jadi dimana setiap komponen-komponen di dalam graf terhubung dari suatu hutan G merupakan pohon-pohon.

Pohon perentang

Pohon perentang suatu graf terhubung G adalah suatu subgraf G yang merupakan pohon dan memuat semua titik dalam G . (Jong Jek Siang, 2009: 286)

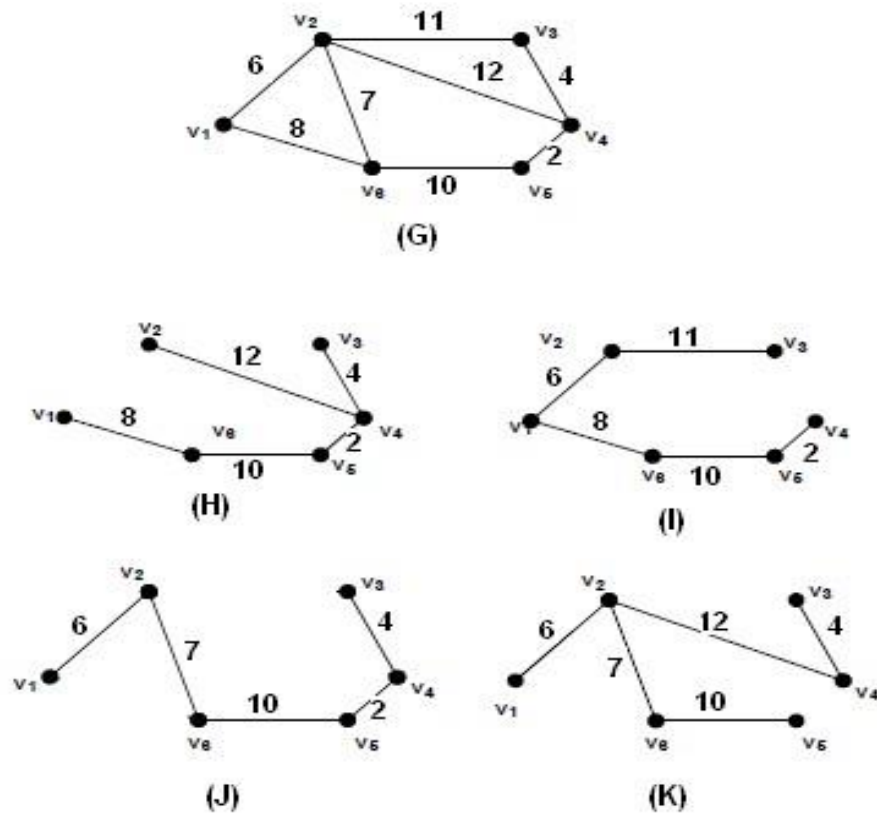
Teorema: Graf sederhana terhubung jika dan hanya jika memiliki sebuah *spanning tree*.

Bukti. (i) Misalkan graf sederhana G memiliki sebuah *spanning tree* T . T mengandung setiap titik dari G . Selanjutnya, ada sebuah *path* di T diantara dua titik, karena T adalah subgraf dari G , ada sebuah *path* di G diantara dua titik tersebut. Oleh karena itu G terhubung.

(ii) Misalkan G terhubung. Jika G bukan sebuah *tree*, mengandung sirkuit sederhana. Hapus sebuah sisi dari salah satu sirkuit sederhana tersebut. Hasilnya, subgraf berkurang satu sisi tetapi masih terhubung dengan semua titik dari G dan terhubung. Jika subgraf bukan *tree* yang memiliki sebuah sirkuit sederhana seperti sebelumnya, hapus sebuah sisi pada sirkuit sederhana tersebut. Ulangi proses ini sampai tidak ada sirkuit sederhana, ini dimungkinkan karena hanya ada berhingga sisi pada graf. Proses berakhir ketika tidak ada sirkuit tersisa. Sebuah *tree* terbentuk selama graf tetap terhubung sama halnya sisi yang dihapus. *Tree* ini merupakan *spanning tree* selama memuat tiap-tiap titik dari G . □

Pohon Perentang Minimum

Suatu pohon perentang minimum dari G adalah pohon rentang yang jumlah bobot totalnya adalah sekecil mungkin. (Samuel Wibisono, 2008: 161). Contoh 2:



Gambar 2 Hutan. (G) graf berbobot, (H,I,J,K) pohon perentang, (I) pohon perentang minimum.

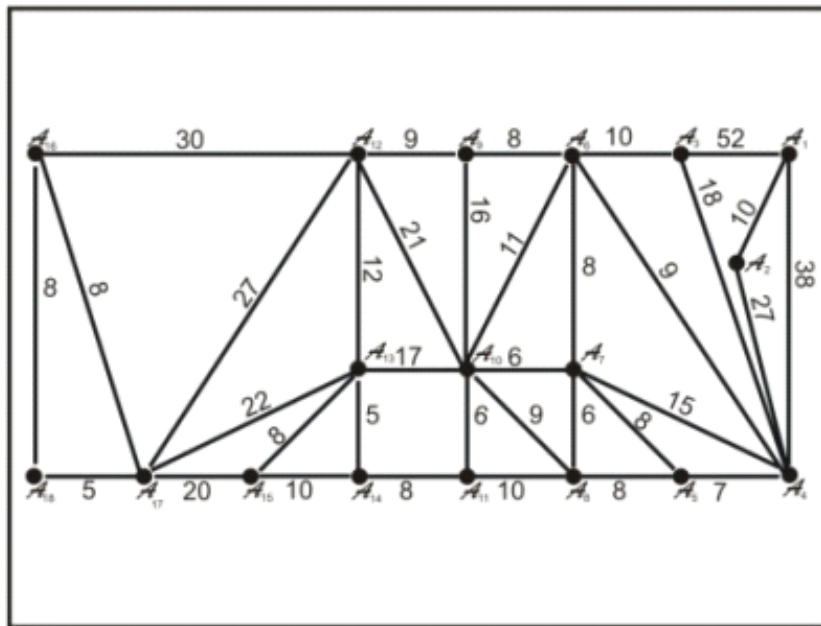
Algoritma Minimum Spanning Tree

Algoritma Boruvka

Algoritma Boruvka adalah algoritma pertama untuk mencari pohon merentang minimum dari suatu graf. Algoritma Boruvka ditemukan oleh Otakar Boruvka pada tahun 1926. (Kenneth H. Rosen, 1999: 483) langkah-langkah mencari pohon perentang minimum dengan algoritma boruvka adalah sebagai berikut.

1. Dibuat hutan F yang terdiri dari semua titik pada G tanpa garis (pohon semu).
2. Setiap pohon pada F ditentukan sisi dengan bobot minimum yang *incident* dengan pohon tersebut dan tidak membentuk *cycle* sehingga pohon tersebut menjadi pohon yang lebih besar. Jika ada dua garis yang sama, pilih salah satu.
3. Langkah 2 diulang sampai semua titik pada F terhubung.

Contoh penerapan graf pada jaringan komputer kampus UII.



Gambar 3 Graf kampus.

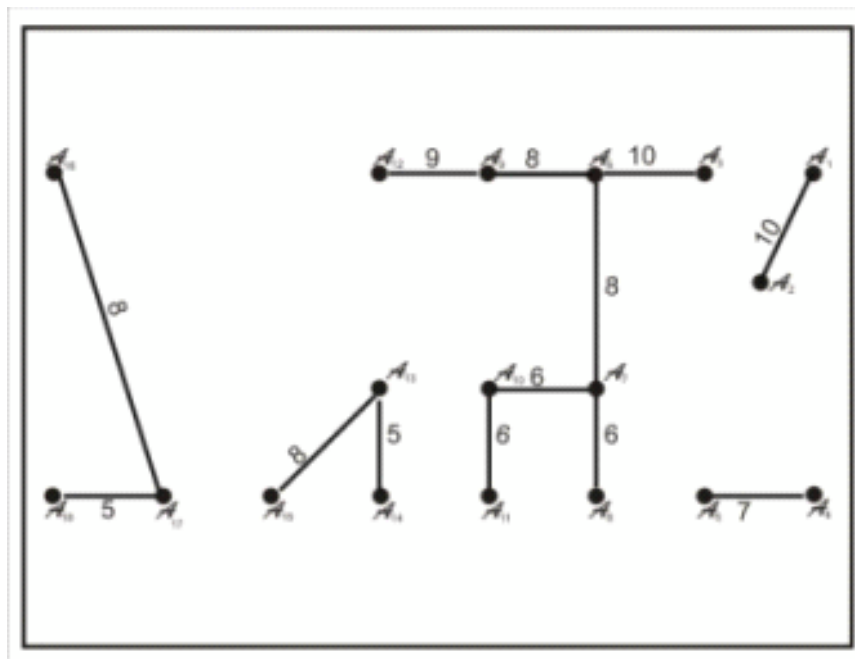
1. Dibuat hutan F yang terdiri dari semua titik. (Tanpa garis).
2. Setiap pohon (titik) di F ditentukan sisi minimum yang *incident*, yaitu:

$$(A1-A2), (A2-A1), (A3-A6), (A4-A5), (A5-A4), (A6-A7), (A7-A8),$$

$$(A8-A7), (A9-A6), (A10-A7), (A11-A10), (A12-A9), (A13-A14), (A14-A13),$$

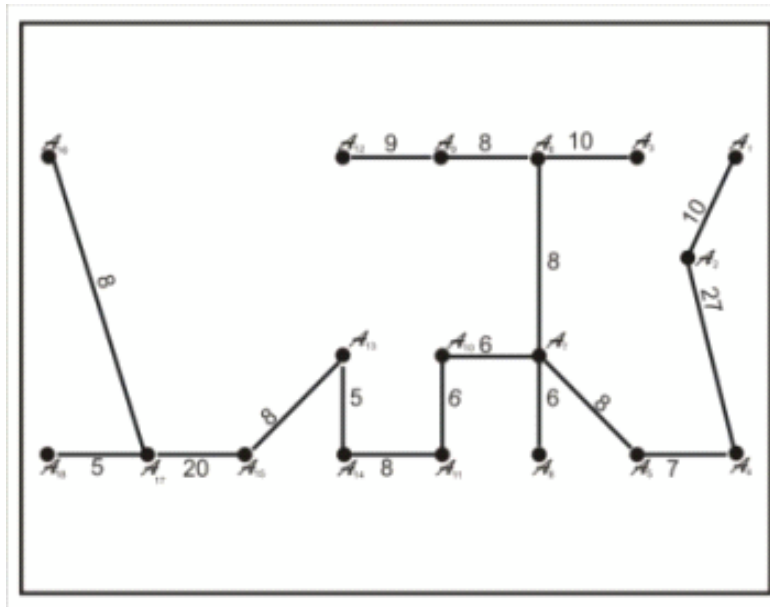
$$(A15-A13), (A16-A17), (A17-A18), (A18-A17)$$

Sehingga didapat hutan seperti di bawah.



Gambar 4 Hutan F langkah 2 algoritma boruvka.

3. Langkah 2 diulang sampai semua titik terhubung.



Gambar 5 Pohon perentang minimum algoritma borovka.

Jumlah bobot total dari pohon perentang minimum yang dihasilkan dengan algoritma burovka adalah 159.

Algoritma Kruskal

Algoritma Kruskal adalah suatu algoritma di dalam teori graf yang digunakan untuk mencari pohon merentang minimum di dalam graf berbobot terhubung secara berurutan dari sisi yang berbobot kecil sampai yang berbobot besar hingga tidak berbentuk sirkel. Algoritma ini pertama kali muncul pada tahun 1956 dalam sebuah tulisan yang ditulis oleh Joseph Kruskal. Langkah-langkah mencari pohon perentang minimum dengan algoritma kruskal adalah sebagai berikut. (Jong Jek Siang, 2009:294)

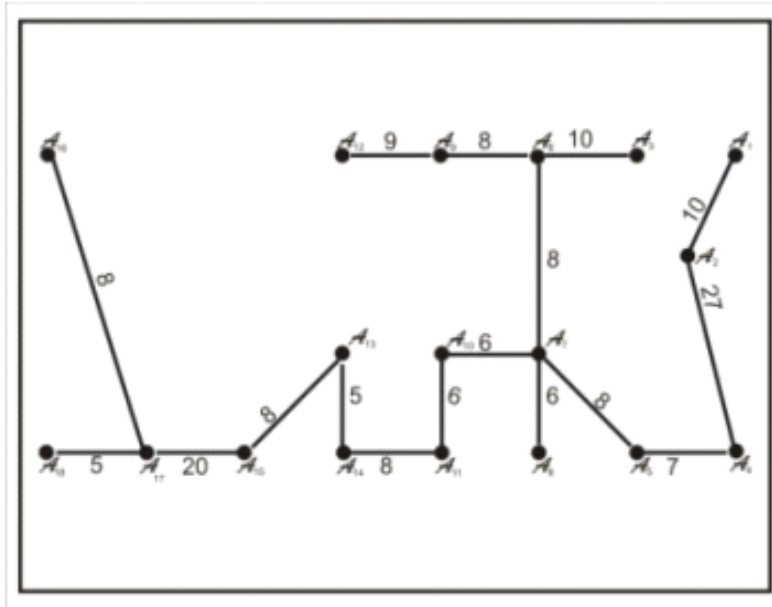
1. Buat hutan F yang terdiri dari semua titik pada graf G_u tanpa garis.
2. Urutkan bobot garis pada G_u dari yang terkecil sampai yang terbesar. Namakan urutan ini dengan W .
3. Pilih garis-garis dengan bobot terkecil dari W untuk menghubungkan setiap dua pohon pada F sehingga membentuk pohon yang lebih besar.
4. Ulangi langkah 3 hingga F membentuk sebuah pohon perentang yang menjangkau semua titik.

Contoh penerapan pada graf kampus UII di bawah ini:

1. Buat Hutan F
2. Urutkan bobot garis pada G dari yang terkecil sampai yang terbesar dalam sebuah himpunan W . Sehingga didapat himpunan $W = \{5,6,7,8,9,10,11,12,15,16,17,18,20, 21,22,27,30,38,52\}$
3. Pilih garis-garis dengan bobot terkecil dari W untuk menghubungkan setiap dua pohon pada F . ($A_{13} - A_{14}$ dan $A_{17} - A_{18}$.)
4. Ulangi langkah 3 hingga semua titik terhubung.
 - Bobot terkecil ($A_7 - A_8, A_7 - A_{10}$, dan $A_{10} - A_{11}$).
 - Bobot terkecil ($A_4 - A_5$)
 - Bobot terkecil ($A_5 - A_7, A_6 - A_7, A_6 - A_9, A_{11} - A_{14}, A_{13} - A_{15}$, dan $A_{16} - A_{17}$)
 - Bobot terkecil ($A_9 - A_{12}$)

- Bobot terkecil ($A_1 - A_2$ dan $A_3 - A_6$)
- Bobot terkecil ($A_{15} - A_{17}$)
- Bobot terkecil ($A_2 - A_4$)

Karena semua titik sudah terhubung, langkah dihentikan. Sehingga didapat pohon merentang minimum seperti di bawah ini:



Gambar 6 Pohon perentang minimum algoritma Kruskal.

Jumlah bobot total dari pohon perentang minimum yang dihasilkan dengan Algoritma Kruskal adalah 159.

Algoritma prim

Algoritma Prim ditemukan oleh Robert C.Prim pada tahun 1957. Algoritma ini membentuk pohon perentang minimum langkah per langkah. Pada tiap langkah diambil sisi yang paling minimum. (Jong Jek Siang, 2002:269) Langkah-langkahnya sebagai berikut:

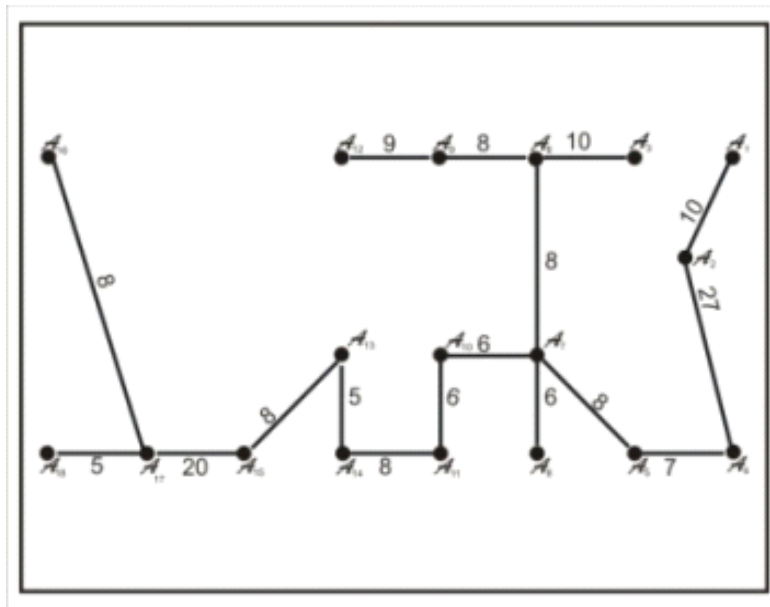
1. Menentukan satu titik pada G sebagai pohon semu T .
2. Pilih salah satu titik lain dengan jarak paling minimum dari T dan tidak membentuk *cycle* sehingga T menjadi pohon yang lebih besar. Jika ada jarak yang sama, pilih salah satu.
3. Langkah ke 2 diulang hingga semua titik pada G terhubung.

Contoh penerapan pada graf kampus UII di bawah ini:

1. Menentukan satu titik pada G sebagai pohon semu T yaitu A_1
2. Pilih salah satu titik lain dengan jarak paling minimum dari T dan tidak membentuk *cycle*.
3. Karena belum terhubung maka langkah 2 diulang.
 - Titik terdekat dari F adalah A_2
 - Titik terdekat adalah A_4
 - Titik terdekat adalah A_5
 - Titik terdekat adalah A_7
 - Titik terdekat adalah A_8
 - Titik terdekat adalah A_{10}
 - Titik terdekat adalah A_{11}
 - Titik terdekat adalah A_6

- Titik terdekat adalah A_9
- Titik terdekat adalah A_{14}
- Titik terdekat adalah A_{13}
- Titik terdekat adalah A_{15}
- Titik terdekat adalah A_{12}
- Titik terdekat adalah A_3
- Titik terdekat adalah A_{17}
- Titik terdekat adalah A_{18}
- Titik terdekat adalah A_{16}

Sehingga didapat pohon perentang minimum.



Gambar 7 Pohon perentang minimum Algorithma Prim.

Jumlah bobot total dari pohon perentang minimum yang dihasilkan dengan algoritma burovka adalah 159.

Algoritma Solin

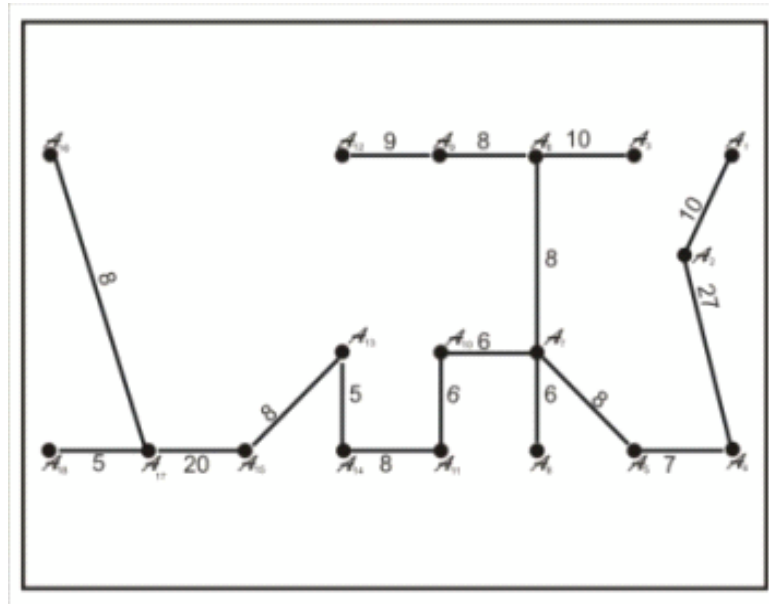
Algoritma Solin ditemukan oleh Sollin pada tahun 1960. Algoritma Solin merupakan pohon perentang minimum dengan cara melakukan penghapusan sisi-sisi yang tidak menyebabkan graf menjadi tidak terhubung atau membentuk sirkuit. Penghapusan tersebut dimulai dari sisi yang memiliki bobot terbesar hingga terkecil. (CharlesWright, 1985:404). Langkah-langkah mencari pohon perentang minimum dengan Algoritma Boruvka sebagai berikut:

1. Urutkan sisi-sisi pada graf berdasarkan bobotnya dari besar ke yang kecil.
2. Lakukan penghapusan setiap sisi yang tidak menyebabkan graf menjadi tidak terhubung.
3. Ulangi langkah kedua hingga diperoleh pohon perentang minimum.

Contoh penerapan pada graf kampus UII di bawah ini:

1. Urutkan sisi-sisi pada graf berdasarkan bobotnya dari besar ke yang kecil. (52,38,30,27,22,21,20,18,17,16,15,12,11,10,9,8,7,6,5)
2. Lakukan penghapusan setiap sisi yang tidak menyebabkan graf menjadi tidak terhubung. sisi $A_1 - A_3$

3. Sisi $(A_1 - A_4), (A_{12} - A_{16}), (A_{12} - A_{16}), (A_{13} - A_{17}), (A_{10} - A_{12}), (A_3 - A_4),$
 $(A_{10} - A_{13}), (A_9 - A_{10}),$
 $(A_4 - A_7), (A_{12} - A_{13}), (A_6 - A_{10}), (A_8 - A_{11}), (A_{14} - A_{15}), (A_4 - A_6), (A_8 - A_{10}), (A_5 - A_8),$
 $(A_{16} - A_{18})$
4. Semua titik sudah terhubung maka langkah dihentikan hingga dapat pohon perentang minimum



Gambar 8 Pohon perentang minimum Algoritma Solin.

Jumlah bobot total dari pohon perentang minimum yang dihasilkan dengan Algoritma Burovka adalah 159. Dari keempat algoritma tersebut di atas memiliki hasil pohon perentang minimum yang sama yaitu dengan panjang total 159, yang berarti panjang jalur 1.590 meter

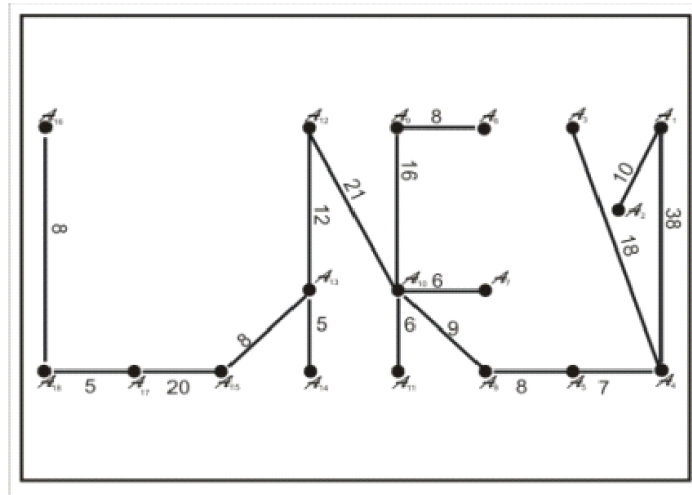
Aplikasi Graf Pada Jaringan Kabel FO Kampus UII

Graf kampus

Graf kampus UII adalah graf yang menggambarkan keterhubungan setiap gedung yang ada di kampus UII melalui jalan-jalan kampus. Jalan-jalan kampus dan jalur khusus kabel merupakan sebuah garis, sedangkan panel-panel yang ada di setiap gedung merupakan titiknya, sehingga dalam graf kampus UII terpadu tersebut memiliki 18 titik.

Graf jaringan Kabel FO UII

Graf jaringan kabel FO UII dapat dibuat dengan bantuan denah persebaran jalur kabel FO UII. Panjang kabel yang dibutuhkan untuk membangun jaringan kabel FO kampus terpadu UII bisa diketahui dengan bantuan graf berbobot dari pohon berakar yang terbentuk dari Graf Jaringan Kabel FO UII.



Gambar 9 Graf jaringan kabel FO UII.

Total bobot T_U adalah 205, yang berarti bahwa panjang total jalur kabel FO UII yang sudah ada saat ini kurang lebih mencapai 2.050 meter dengan panjang total kabel 6.120 meter.

Kesimpulan

Dari hasil penelitian menggunakan Algoritma Kruskal, Prim, Boruvka, dan Algoritma Solin pada graf kampus UII menghasilkan pohon perentang minimum (*minimum spanning tree*) yang sama yaitu

- Panjang total jalur jaringan kabel FO adalah 1.590 meter.
- Panjang total kabel FO yang dibutuhkan adalah 4.700 meter.

Hasil penelitian tersebut lebih minimum 23,2% dibandingkan kenyataan yang terjadi pada jaringan kabel FO kampus UII saat ini, yaitu

- Panjang total jalur jaringan kabel FO adalah 2.050 meter.
- Panjang total kabel FO yang dibutuhkan adalah 6.120 meter.

Referensi

- [1] Abdusakir. 2009. *Teori Graf Topik Dasar Untuk Tugas Akhir/Skripsi*. Malang: UIN Malang
- [2] Lipschutz, Seymour. 2008. *Matematika Diskrit 1*. Jakarta: Erlangga
- [3] Madcoms. 2013. *Cepat dan Mudah Membangun Sistem Jaringan Komputer*. Yogyakarta: Andi.
- [4] Rosen, Kenneth. H. 1999. *Discrete Mathematics and Its Applications*. New York: McGraw-Hill.
- [5] Siang, Jong Lek. 2009. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi
- [6] Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. Yogyakarta: Andi.
- [7] Tim Penulis, Wahana Komputer. 2003. *Konsep Jaringan Komputer dan Pengembangannya*. Jakarta: Salemba Infotek.
- [8] Wibisono, Samuel. 2008. *Matematika Diskrit*. Yogyakarta: Graha Ilmu.
- [9] Wright, Charles R.B. 1985. *Discrete Mathematics*. London: Prentice-Hall.