

PENERAPAN ALGORITMA GREEDY DALAM MENENTUKAN *MINIMUM SPANNING TREES* PADA OPTIMISASI JARINGAN LISTRIK JALA

Didiharyono¹, Siti Soraya²

Universitas Andi Djemma Palopo¹, STMIK BUMIGORA Mataram

E-mail: didi.unanda@gmail.com, siti.soraya@stmikbumigora.ac.id

Abstract

This article discusses the applied of greedy algorithm principle in finding the optimum solution in determine minimum spanning tree on graph. Graph theory is one of the studies in discrete mathematics that are widely applied in various scope. This article is a literature study and applied of nets electricity network optimization using Prims algorithm and Kruskal algorithm. Network Nets System is one type of electrical network system construction. Based on results of the study and discussion can be concluded that the application of greedy algorithm using Prims algorithm and Kruskal algorithm in determine minimum spanning tree on its principle is the same. However, after a comparison between the two algorithms we consider that the ideal algorithm used to optimize the nets electric network is the Kruskal algorithm because in the case of the electric network has few sides and many vertices.

Keywords: *Greedy Algorithm, Minimum Spanning Trees and Nets Electricity Network*

Abstrak

Artikel ini membahas penerapan prinsip algoritma greedy dalam menemukan solusi optimum pada penentuan *minimum spanning tree* (pohon merentang minimum) pada graf. Teori graf merupakan salah satu kajian dalam matematika diskrit yang banyak diterapkan dalam berbagai kelimuan. Artikel ini merupakan studi pustaka dan aplikasinya pengoptimalan jaringan listrik jala dengan menggunakan algoritma Prims dan algoritma Kruskal. Sistem jaringan jala sendiri merupakan salah satu jenis konstruksi sistem jaringan listrik. Berdasarkan hasil kajian dan pembahasan dapat disimpulkan bahwa penerapan algoritma greedy dengan menggunakan algoritma Prims dan algoritma Kruskal dalam menentukan *minimum spanning tree* pada prinsipsinya adalah sama. Namun, setelah dilakukan perbandingan antara kedua algoritma tersebut maka kami menganggap bahwa algoritma yang ideal digunakan untuk mengoptimalkan jaringan listrik khusus jaringan listrik jala adalah algoritma Kruskal karena pada kasus jaringan listrik tersebut memiliki jumlah sisi yang sedikit dan memiliki banyak simpul.

Kata Kunci: *Algoritma greedy, Minimum Spanning Trees dan Jaringan Listrik Jala*

I. PENDAHULUAN

Algoritma *greedy* adalah salah satu bentuk algoritma yang digunakan untuk memecahkan persoalan optimasi. Beberapa persoalan optimasi yang akan dibahas dalam algoritma *greedy* yaitu berkaitan dengan maksimasi dan minimasi. Dalam artian bahwa metode *greedy* bermaksud menemukan solusi terbaik yaitu solusi yang bernilai minimum atau maksimum dari beberapa alternatif solusi yang ada. Algoritma *greedy* adalah algoritma

yang memecahkan masalah langkah demi langkah (*step by step*), pada setiap langkah membuat pilihan optimum (*local optimum*), pada setiap langkah dengan harapan bahwa langkah berikutnya mengarah ke solusi optimum global (*global optimum*)[1].

Algoritma *greedy* membuat keputusan berdasarkan pilihan yang ada sekarang, tidak melihat pilihan yang akan muncul kemudian atau di masa yang akan datang. Karena itulah algoritma *greedy* dikategorikan dalam

algoritma yang digunakan untuk menentukan solusi optimum dan tidak dapat diulang karena keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah-langkah selanjutnya. Padahal dalam permasalahan optimasi terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah untuk menemukan solusi yang terbaik.

Terkadang algoritma *greedy* mengambil keputusan yang diambil terlalu dini tanpa melihat yang akan ditemui berikutnya sehingga menimbulkan masalah yang disebut “*good next move, bad overall move*”. Melihat kelemahan yang dimiliki, solusi optimum global yang didapatkan belum tentu merupakan solusi optimum (terbaik), tetapi sub-optimum atau *pseudo-optimum*. Karena algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada. Sehingga, algoritma ini tetap menjadi pilihan utama untuk menyelesaikan permasalahan yang ada dan dapat memberikan solusi optimum yang diinginkan serta hasil yang diberikan masih merupakan solusi yang terbaik (*feasible solution*).

Algoritma yang termasuk ke dalam tipe algoritma *greedy* antara lain kode *Huffman*, algoritma *Dijkstra*, algoritma *Prims* dan algoritma *Kruskal*, yang keempatnya digunakan dalam menyelesaikan permasalahan dalam berbagai bidang termasuk dalam materi graf [9].

Penentuan *minimum spanning trees* atau pohon merentang minimum merupakan suatu masalah klasik dalam ilmu komputasi. Algoritma yang akan digunakan untuk mencari *minimum spanning trees* adalah algoritma *Prim's* dan algoritma *Kruskal*.

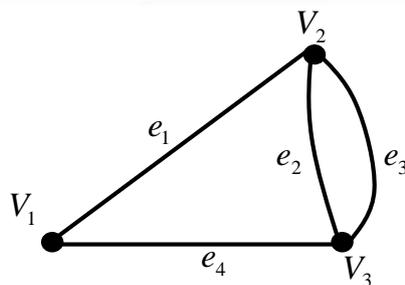
Algoritma *Prims* dan *Kruskal* termasuk bagian dari algoritma *greedy* yang digunakan dalam menyelesaikan masalah dan pada setiap langkah memiliki solusi optimal lokal yang diharapkan akan membentuk solusi optimal global [2]. Algoritma *Prims* dan *Kruskal* dapat

digunakan dalam mencari solusi optimum pada penentuan *minimum spanning trees* dalam graf. Dalam graf ini, digunakan untuk mempresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf dinyatakan berupa objek sebagai noktah atau titik, sedangkan hubungan antara objek-objek dinyatakan dalam bentuk garis.

Dewasa ini aplikasi graf telah banyak digunakan untuk merepresentasikan permasalahan yang ada agar lebih mudah dipecahkan. Dalam bidang kimia, graf digunakan dalam memodelkan molekul senyawa karbon. Bidang teknik elektro graf digunakan dalam perancangan *integrated circuit* serta masalah kemacetan lalu lintas dapat diselesaikan dengan memodelkan jalan raya dengan menggunakan graf. Penggunaan teori graf banyak memberikan solusi untuk menyelesaikan permasalahan yang terjadi di dalam masyarakat. Ilmu graf terapan tersebut terus berkembang, bahkan kini graf juga dapat digunakan untuk optimisasi dalam berbagai bidang. Adapun tujuan dari artikel ini adalah untuk menentukan penerapan dari algoritma *greedy* dalam menentukan *minimum spanning trees* pada optimisasi jaringan listrik jala yaitu dengan menggunakan algoritma *Prims* dan algoritma *Kruskal* yang merupakan bagian dari algoritma *greedy*.

1.1 Teori Graf

Graf didefinisikan sebagai $G = (V, E)$ dengan V merupakan himpunan tidak kosong dari simpul-simpul (*vertices/ node*) digambarkan dalam titik-titik, dan E adalah himpunan sisi-sisi (*edges/ arcs*) digambarkan dalam garis-garis yang menghubungkan sepasang simpul. Graf dapat dikatakan sebagai kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi. Perhatikan Gambar 1 berikut:



Gambar 1. Graf G

Pada Gambar 1 di atas, graf G terdiri dari himpunan V dan E yaitu:

$$V = \{V_1, V_2, V_3\}$$

$$E = \{e_1, e_2, e_3, e_4\} \quad \text{atau}$$

$$\{(V_1, V_2), (V_2, V_3), (V_2, V_3), (V_1, V_3)\}.$$

Aplikasi graf sangat luas. Graf dipakai dalam berbagai disiplin ilmu maupun dalam kehidupan sehari-hari. Penggunaan graf di berbagai bidang tersebut adalah untuk memodelkan persoalan. Beberapa terminologi dasar yang harus diketahui:

Definisi 1. Misalkan $G = (V, E)$ adalah graf dan u, v di G . Graf G disebut graf terhubung (*connected graph*) jika $\forall u, v \in V(G)$ terdapat lintasan di G yang menghubungkan u dan v .

Definisi 2. Misalkan $G = (V, E)$ adalah graf. Sembarang fungsi bertipe $w: E \rightarrow \mathbb{N}$ dinamakan fungsi bobot. Graf G , bersama-sama dengan fungsi $w: E \rightarrow \mathbb{N}$, disebut graf berbobot (*weighted graph*)

Definisi 3. Misalkan $G = (V, E)$ adalah graf berbobot dan terhubung, maka G disebut graf terhubung berbobot.

Definisi 4. Misalkan $G = (V, E)$ adalah graf terhubung tak bera

rah dan $T = (V_1, E_1)$ adalah graf pohon. Graf T disebut pohon perentang (*spanning tree*) dari G , jika $V_1(T) = V(G)$ dan $E_1(T) \subseteq E(G)$

Definisi 5. Misalkan $G = (V, E)$ adalah graf terhubung berbobot. Misalkan T adalah suatu pohon perentang dari G . Nilai

$$w(T) = \sum_{e \in T} w(e)$$

Jumlah bobot dari sisi-sisi di T , disebut bobot dari pohon rentangan T .

1.2 Konsep Tree (Pohon)

Teori graf merupakan salah satu cabang matematika yang memiliki banyak penerapan untuk mencari solusi dari permasalahan diskrit yang terjadi di kehidupan sehari-hari. Untuk mencari solusi tersebut, di dalam graf terdapat banyak konsep. Salah satu konsep di antaranya adalah konsep pohon (*tree*). Pohon adalah graf tak berarah terhubung yang tidak mengandung sirkuit.

Salah satu istilah dalam konsep pohon adalah pohon perentang (*spanning tree*). Misalkan $G = (V, E)$ adalah graf terhubung tak berarah yang bukan pohon, berarti pada graf G terdapat beberapa siklus, T dapat diubah menjadi suatu pohon $T = (V_1, E_1)$ dengan cara menghapus sisi-sisi yang membentuk siklus sehingga graf terhubung tidak lagi memuat siklus, graf G menjadi sebuah pohon T yang disebut pohon perentang.

Salah satu jenis graf khusus adalah graf berbobot (*weighted graph*). Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga atau bobot. Pada graf berbobot terhubung, dikenal istilah pohon perentang minimum (*minimum spanning tree*). Jika G adalah graf berbobot, maka bobot pohon perentang T dari G didefinisikan sebagai jumlah bobot semua sisi di T . Di antara semua pohon perentang dari G , pohon perentang yang berbobot minimum disebut pohon perentang minimum.

1.3 Algoritma Greedy untuk Minimum Spanning Trees

Secara jelas, untuk sembarang pohon perentang T dari G memiliki $w(T) \in \mathbb{N}$. Jelas juga bahwa hanya terdapat berhingga pohon perentang T dari G . Karena itu pasti ada satu pohon perentang T , dimana nilai bobot $w(T)$ paling kecil diantara semua pohon perentang dari G .

Definisi 6. Misalkan $G = (V, E)$ adalah graf terhubung berbobot. Suatu pohon perentang T dari G , dengan bobot $w(T)$ adalah terkecil diantara semua pohon perentang dari G , disebut pohon perentang minimum (*minimum spanning trees*) dari G .

Pohon perentang minimum dari suatu graf terhubung berbobot mungkin tidak tunggal. Untuk membentuk suatu

pohon perentang minimum dari sembarang graf terhubung berbobot dapat menggunakan suatu algoritma. Algoritma *greedy* yang membentuk suatu pohon perentang minimum dari sembarang graf terhubung berbobot, yaitu:

a. Algoritma *Prim's*

Algoritma ini ditemukan oleh matematikawan Vojt ch Jarník pada tahun 1930 dan kemudian secara terpisah ditemukan oleh ahli komputer Robert C. Prim pada tahun 1957. Strategi *greedy* yang digunakan pada setiap langkah, pilih sisi dari graf G yang mempunyai bobot minimum dengan syarat sisi tersebut tetap terhubung dengan pohon merentang minimum T yang telah terbentuk.

Pseudo-code algoritmanya adalah:

```

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 rentang 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  $n-2$  do
    Pilih sisi (u,v) dari  $E$  yang bobotnya terkecil namun bersisian
    dengan suatu simpul di dalam  $T$ 
     $T \leftarrow T \cup \{(u,v)\}$ 
  Endfor
    
```

Algoritma *Prim's* untuk suatu Pohon Perentang Minimum.

Input : Graf terhubung $G = (V, E)$, dan dilengkapi fungsi bobot $w: E \rightarrow \mathbb{N}$.

Output: Pohon perentang minimum T dari G .

- 1) Ambil satu titik u sebarang di G , untuk membentuk pohon T .
- 2) T akan dibentuk dengan mengambil sisi di $E(G)$ yang berbobot minimum.

- 3) Pilih sisi $\{u, v\} \in E(G) \setminus E(T)$ yang bersisian dengan titik di $E(T)$, memiliki bobot minimum, dan tidak membentuk siklus. Tambahkan $\{u, v\}$ ke dalam $E(T)$.
- 4) Ulangi langkah (2) sebanyak $(n - 2)$ kali sampai semua titik di $V(G)$ termuat dalam pohon. Jumlah langkah seluruhnya pada algoritma *Prim's* adalah $1 + n - 2 = n - 1$, yaitu sebanyak jumlah sisi di dalam pohon perentang T dengan n titik.

b. Algoritma *Kruskal*

Algoritma ini pertama kali muncul pada tahun 1956 dalam sebuah tulisan yang ditulis oleh Joseph Kruskal. Strategi *greedy* yang digunakan: Pada

setiap langkah, pilih sisi dari graf G yang mempunyai bobot minimum tetapi sisi tersebut tidak membentuk sirkuit di T . *Pseudo-code* algoritmanya adalah:

```
procedure Kruskal (input  $G$  : graf, output  $T$  : pohon)
{ Membentuk pohon merentang minimum  $T$  dari graf terhubung  $G$ .
  Masukkan: 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 bobotnya }
 $T \leftarrow \{\}$ 
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
     $T \leftarrow T \cup \{(u,v)\}$ 
  endif
endfor
```

Algoritma Kruskal untuk suatu Pohon Perentang Minimum.

Input : Graf terhubung $G = (V, E)$ dengan $|V| = n$, dan dilengkapi fungsi bobot $w: E \rightarrow \mathbb{N}$.

Output : Pohon perentang minimum T dari G .

- 1) Bentuk pohon T hanya terdiri dari seluruh titik $V(G)$, tanpa memasukkan sisi $E(G)$
- 2) Semua sisi di $E(G)$ diurutkan berdasarkan ukuran bobot mulai bobot terkecil sampai terbesar.
- 3) Pohon T akan dibentuk dengan mengambil sisi $\{u,v\} \in E(G) \setminus E(T)$ yang memiliki bobot minimum sesuai ukuran bobot, dan tidak membentuk siklus di T . Tambahkan $\{u, v\}$ ke dalam $E(T)$.
- 4) Ulangi langkah (2) sebanyak $(n - 1)$ kali sampai semua titik di $V(G)$ termuat dalam pohon.

salah satu jenis konstruksi sistem jaringan listrik. Adapun langkah-langkah algoritma yang digunakan dalam artikel ini adalah sebagai berikut:

- a. Permasalahan yang dibahas dalam artikel ini adalah masalah yang berkaitan dengan optimalisasi jaringan listrik. Itu berarti bahwa jaringan listrik yang diduga belum optimal, maka perlu dilakukan penelitian lebih lanjut, yaitu dengan cara melakukan pengukuran jarak antar rumah, antar tiang listrik, antara rumah dan tiang listrik.
- b. Setelah dilakukan pengukuran, data yang diperoleh hasil pengukuran direpresentasikan dalam graf yang mana graf hasil representasi tersebut siap untuk dianalisis.
- c. Algoritma yang digunakan dalam menganalisis hasil pengukuran tersebut dengan menggunakan algoritma *Prims* dan algoritma *Kruskal*
- d. Mendefinisikan masalah *minimum spanning trees* yang diselesaikan yaitu dengan membandingkan penyelesaian algoritma *Prims* dan algoritma *Kruskal*. Sehingga, dalam penerapan algoritma *greedy* maka jaringan listrik dapat dioptimalkan.

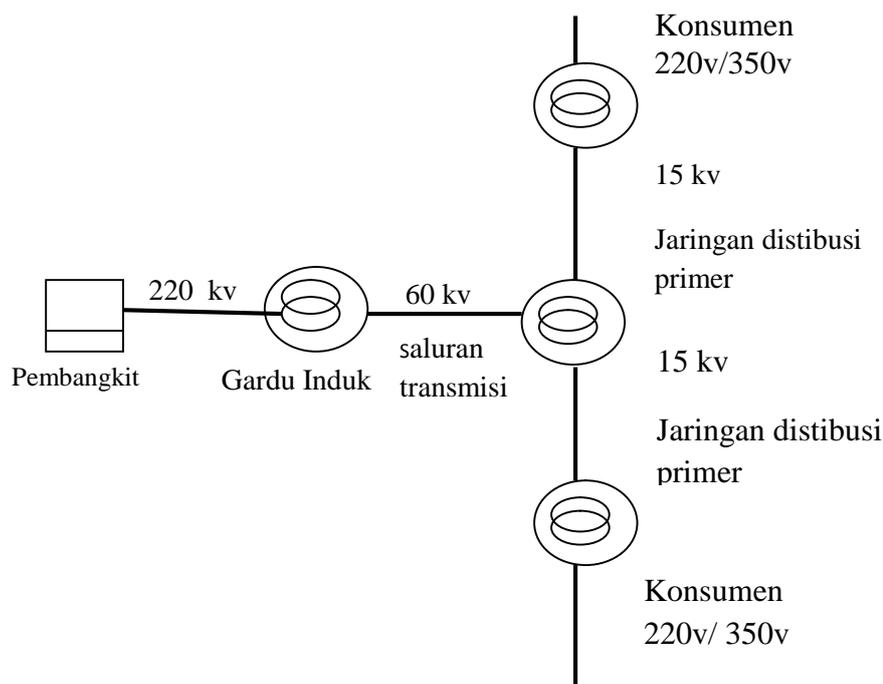
II. METODOLOGI PENELITIAN

Artikel ini merupakan studi pustaka dengan memberikan contoh penentuan optimasi sistem jaringan jala yang merupakan

III. HASIL DAN PEMBAHASAN

Penyaluran atau transmisi energi listrik dari pusat pembangkit listrik dilakukan dengan kabel melalui dua saluran yaitu saluran udara atau saluran bawah tanah. Membandingkan dengan transmisi saluran bawah tanah, transmisi dengan saluran udara memiliki beberapa keuntungan, yaitu [10]: isolasinya lebih mudah pendinginnya

baik, gangguan-gangguan lebih mudah diatasi dengan cepat, dan jauh lebih murah. Gambar 2 menunjukkan diagram penyaluran energi listrik dari pusat pembangkit sampai konsumen. Untuk jaringan distribusi ini kebanyakan menggunakan saluran udara, kecuali dibagian-bagian kota yang padat menggunakan saluran bawah tanah.



Gambar 2. Blok Penyaluran Energi Listrik

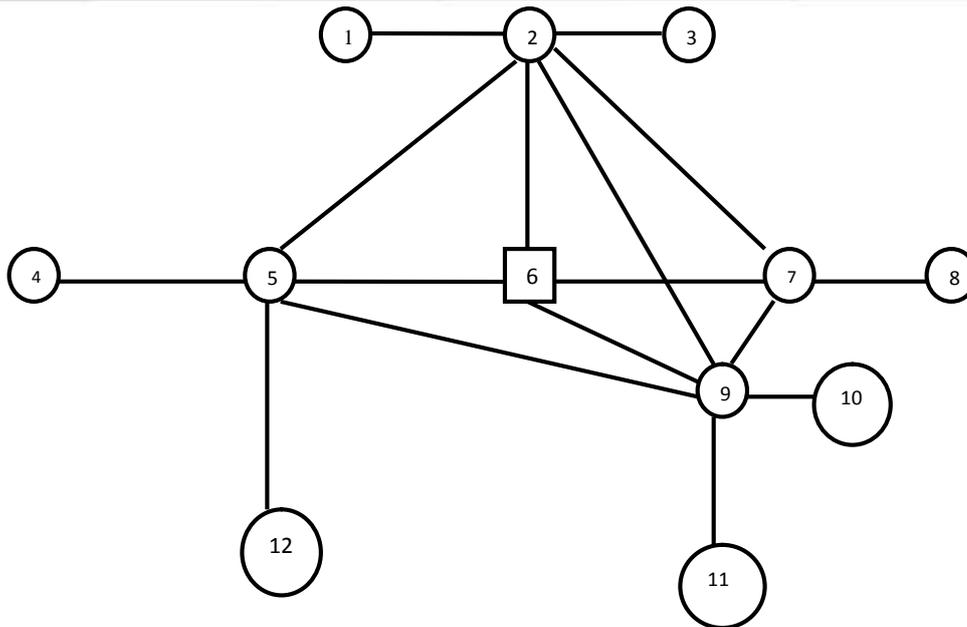
Bila ditinjau dari konstruksi sistem jaringan dibedakan menjadi beberapa jenis antara lain:

- Sistem radial. Sistem jaringan ini biasanya gardu-gardu induk dihubungkan langsung dengan pusat listrik. Gardu-gardu transformatornya dihubungkan langsung dengan salah satu gardu induk. Sistem ini digunakan jika letak gardu-gardu induknya tersebar, saling berjauhan dan jauh dari pusat listrik.
- Sistem jaringan lingkaran. Sistem jaringan ini gardu-gardu induk dihubungkan berderet, sehingga membentuk lingkaran dengan pusat pembangkit. Gardu-gardu

transformatornya juga dihubungkan berderet membentuk lingkaran dengan salah satu gardu induk.

- Sistem jaringan jala. Sistem jaringan ini gardu-gardu induk dihubungkan langsung dengan pusat listrik. Selain itu, gardu induk yang satu juga dihubungkan dengan yang lain.

Berdasarkan konstruksi sistem jaringan listrik, salah satu sistem jaringan listrik adalah sistem jaringan jala yang biasa juga disebut sistem paling handal, dimana gardu-gardu induk dihubungkan langsung dengan pusat listrik.



Gambar 3. Sistem Jaringan Jala

Jaringan listrik dapat dioptimalisasi dengan menerapkan aplikasi algoritma greedy dalam pohon perentang atau *minimum spanning tree* (MST) dalam pembuatan jaringan listrik. Jaringan listrik di representasikan sebagai graf, dimana tiang listrik dan rumah sebagai titik, dan sisi merupakan kabel. Maka dalam

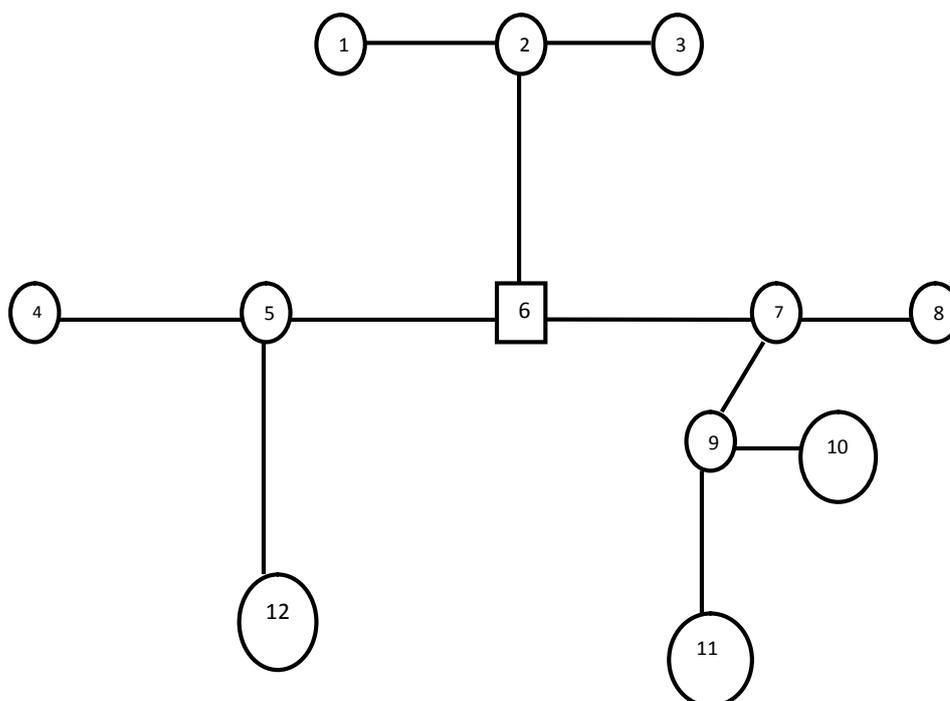
masalah ini, kami menggunakan algoritma *greedy*. Graf sistem jaringan jala tersebut yang akan dicari pohon perentangya menggunakan algoritma *greedy* yaitu algoritma *Kruskal* dan algoritma *Prims* serta melakukan perbandingan antara kedua algoritma tersebut.

Tabel 1. Langkah Pembentukan MST menggunakan algoritma Prims

Langkah	Prims	
	Simpul	Sisi Terbentuk
0	9	-
1	10	(9,10)
2	11	(9,10),(11,9)
3	7	(9,10),(11,9),(9,7)
4	8	(9,10),(11,9),(9,7),(8,7)
5	6	(9,10),(11,9),(9,7),(8,7),(7,6),
6	5	(9,10),(11,9),(9,7),(8,7),(7,6),(6,5)
7	4	(9,10),(11,9),(9,7),(8,7),(7,6),(6,5),(4,5)
8	12	(9,10),(11,9),(9,7),(8,7),(7,6),(6,5),(4,5), (5,12)
9	2	(9,10),(11,9),(9,7),(8,7),(7,6),(6,5),(4,5), (5,12), (6,2)
10	3	(9,10),(11,9),(9,7),(8,7),(7,6),(6,5),(4,5), (5,12), (6,2),(3,2)

11	1	(9,10),(11,9),(9,7),(8,7),(7,6),(6,5),(4,5), (6,2),(3,2),(2,1)	(5,12),
----	---	---	---------

Berdasarkan langkah pembentukan menggunakan algoritma Prims, maka MST untuk pengoptimalan jaringan listrik diperoleh MST



Gambar 4. MST Sistem Jaringan Listrik Berdasarkan Algoritma Prims

Selanjutnya, akan ditunjukkan langkah pembentukan minimum

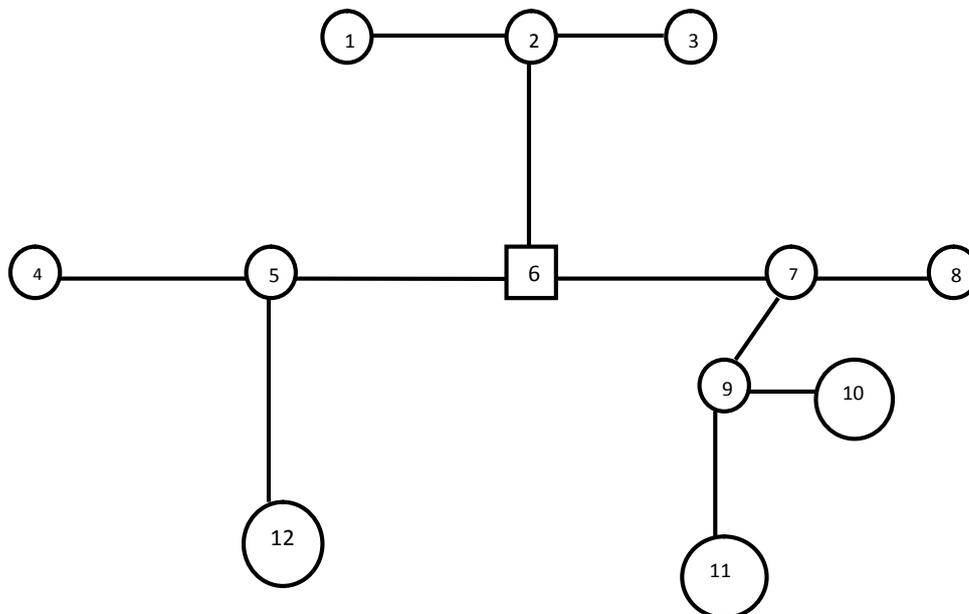
spanning tree dengan menggunakan algoritma *Kruskal*

Tabel 2. Langkah pembentukan MST menggunakan algoritma *Kruskal*

Langkah	Kruskal	
	Sisi	Simpul masuk
1	(9,10)	(9,10)
2	(9,7)	(7,9,10)
3	(9,11)	(7,9,10,11)
4	(2,3)	(2,3,7,9,10,11)
5	(2,1)	(1,2,3,7,9,10,11)
6	(7,8)	(1,2,3,7,8,9,10,11)
7	(2,6)	(1,2,3,6,7,8,9,10,11)

8	(5,4)	(1,2,3,4,5,6,7,8,9,10,11)
9	(5,12)	(1,2,3,4,5,6,7,8,9,10,11,12)
10	(6,5),(6,7)	(1,2,3,4,5,6,7,8,9,10,11,12)

Berdasarkan langkah pembentukan menggunakan algoritma *Kruskal*, maka MST untuk pengoptimalan jaringan listrik diperoleh MST



Gambar 5. MST Sistem Jaringan Listrik Berdasarkan Algoritma Kruskal

Berdasarkan Tabel 1 dan Tabel 2 terlihat bahwa secara umum, algoritma *Kruskal* bisa berjalan hampir sama dibandingkan algoritma *Prims*. Namun jumlah simpul dan jumlah sisi dalam graf juga menentukan algoritma apa yang sebaiknya dipakai karena tiap algoritma memiliki kekuatan tersendiri sesuai dengan spesifikasinya. Ketika graf memiliki sisi berjumlah sedikit namun memiliki sangat banyak simpul maka algoritma *Kruskal* akan lebih cocok diterapkan karena orientasi kerja algoritma ini adalah berdasarkan pada urutan bobot sisi, tidak berdasarkan simpul. Sebaliknya untuk graf lengkap atau yang mendekati lengkap, dimana setiap simpul terhubung dengan semua simpul yang lain algoritma *Prim's* menunjukkan perbedaan

yang cukup besar dibandingkan dengan algoritma *Kruskal*. Algoritma *Prims* lebih berorientasi kepada pencarian simpul, sedangkan algoritma *Kruskal* pada pencarian sisi, dimana sisi-sisi tersebut harus diurutkan.

IV. KESIMPULAN

Berdasarkan hasil kajian dan pembahasan dapat disimpulkan bahwa penerapan algoritma *greedy* dengan menggunakan algoritma *Prim's* dan algoritma *Kruskal* dalam menentukan *minimum spanning tree* pada hakikatnya adalah sama. Namun, setelah dibandingkan antara kedua algoritma tersebut maka kami menganggap bahwa algoritma yang ideal digunakan untuk mengoptimalkan jaringan listrik khusus

jaringan listrik jala adalah algoritma Kruskal karena pada kasus jaringan listrik tersebut memiliki jumlah sisi yang sedikit dan memiliki banyak simpul.

V. DAFTAR PUSTAKA

- [1] Chartrand, Gary dan Zhang, Ping. (2004). *Introduction to Graph Theory*. New York: McGraw-Hill International.
- [2] Emut. (2008). *Aplikasi Algoritma Prime Dalam Menentukan Pohon Pembangkit Minimum Suatu Graf (Studi Kasus)*. Universitas Negeri Yogyakarta.
- [3] Haryono, D., 2014. *Filsafat Matematika*. Bandung: Alfabeta.
- [4] Jatiningih, Trisni. (2010). *Rancang Bangun Aplikasi Minimum Spanning Tree (MST) Menggunakan Algoritma Kruskal*. Naskah Publikasi Jurnal Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (AMIKOM) Yogyakarta.
- [5] Nugraha, D. W. (2011). Aplikasi algoritma prim untuk menentukan minimum spanning tree suatu graf berbobot dengan menggunakan pemrograman berorientasi objek. In *FORISTEK: Forum Teknik Elektro dan Teknologi Informasi* (Vol. 1, No. 2).
- [6] Nugroho, Didit Budi. (2008). *MX 324 Pengantar Teori Graf*. Universitas Kristen Satya Wacana.
- [7] Nugroho, Dwi Satio. (2015). Penerapan Algoritma Reverse Delete dalam Menentukan Minimum Spanning Tree Obyek Wisata Di Kota Yogyakarta. *Universitas Islam Negeri Sunan Kalijaga*.
- [8] Rahmawati, Dian, and Ade Candra. (2013). Implementasi algoritma greedy untuk menyelesaikan masalah knapsack problem. *Jurnal Ilmiah SAINTIKOM* 12.
- [9] Rosen, Kenneth. (2012). *Discrete Mathematics and Its Applications*. Seventh Edition. New York : McGraw-Hill Companies, Inc
- [10] Syaputra, Aidil. (2012). *Aplikasi Pohon Merentang (Spanning Tree) Dalam Pengoptimalan Jaringan Listrik*. Makalah IF2091 Struktur Diskrit, Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.