APLIKASI POHON MERENTANG MINIMUM UNTUK MENENTUKAN JARINGAN DISTRIBUSI LISTRIK

Sidik Soleman (13508101)

Program Studi Teknik Informatika, STEI ITB
Jalan Ganesha 10 Bandung
e-mail: sidik_324@students.itb.ac.id

ABSTRAK

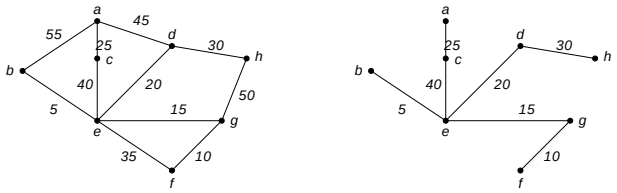
Makalah ini akan membahas mengenai pohon merentang minimum dan aplikasinya dalam menentukan jalur distribusi listrik. Pohon merentang minimum dapat dibuat dengan beberapa algoritma dari sebuah graf. Algoritma tersebut antara lain Algoritma Prim, Algoritma Kruskal, Algortima Boruvka, dan Algoritma Sollin atau dengan kombinasi beberapa algoritma tersebut. Algoritma yang paling sering digunakan adalah Algoritma Prim dan Kruskal. Pohon merentang cocok untuk memodelkan persoalan yang berkaitan antara objek yang satu dengan objek lain dengan menganalisis hubungan antarobjek tersebut. Pohon merentang dapat diaplikasikan dalam persoalan mencari jarak minimum sehingga dari satu titik hanya memiliki satu jalur ke titik lain yang memiliki banyak jalur. Persoalan distribusi listrik dapat dimodelkan dengan pohon merentang minimum. Pada proses distribusi listrik, listrik harus didistribusikan dari suatu pembangkit listrik ke gardu-gardu lainya dan kemudian ke gardu yang lebih kecil dan sampai pada akhirnya listrik disampaikan ke rumah-rumah warga. Pemanfaatan pohon merentang minimum pada distribusi listrik akan menghemat biaya pembuatan jaringan listrik tersebut karena kawat yang digunakan sebagai media penyalur listrik digunakan dengan jumlah yang paling minimum.

Kata kunci: Algortima Prim, Algoritma Kruskal, Algoritma Sollin, Algortima Boruvka, Pohon merentang minimum, distribusi listrik.

1. PENDAHULUAN

Pohon merupaka graf berarah yang tidak memiliki sirkuit. Pohon merentang graf yang seluruh sirkuitnya telah dibuang sehingga masing-masing simpul hanya memiliki satu jalur ke simpul yang lain. Karena itu pohon merentang graf G adalah pohon yang memiliki simpul sebanyak simpul graf G dan sisi-sisi pohon tersebut merupakan himpunan bagian sisi-sisi graf G. Sedangkan pohon merentang minimum adalah graf yang seluruh sirkuitnya telah dihilangkan dan masing-masing simpul hanya memiliki satu jalur ke simpul lainya dengan bobot yang paling minimum. Graf ini biasanya merupakan graf berbobot.

Pohon merentang minimum dapat diperoleh dengan beberapa algoritma antara lain algoritma prim, kruskal, boruvka, dan sollin. Aplikasi pohon merentang sangat banyak terutama untuk masalah pembuatan atau perawatan jaringan agar mengeluarkan biaya yang minimum. Salah satunya adalah distribusi listrik. Jaringan distribusi listrik dituntut untuk mampu mengalirkan listrik dari pembangkit listrik ke rumah-rumah pelanggan melalui gardu-gardu.



Gambar 1. Graf G dan Pohon Merentang G

Pada gambar di atas yaitu gambar graf G menunjukkan dengan salah satu algoritma atau kombinasi algoritma-algoritma tersebut dapat diperoleh pohon merentang G yang minimum.

Sisi-sisi pada pohon merentang disebut pohon, sedangkan sisi-sisi graf yang tidak ada pada pohon merentang disebut tali-hubung. Jika tali-hubung ini dikembalikan lagi ke pohon merentang tali-hubung ini akan membuat sirkuit pada pada graf tersebut sehingga pohon tersebut telah berubah menjadi graf.

Simpul-simpul pada graf tersebut dapat diilustrasikan sebagai gardu-gardu penyalur listrik yang memiliki anak berupa jaringan gardu dibawahnya atau rumah-rumah.

Sedangkan akar dari pohon merentang tersebut merupakan pembangkit listrik.

2. ALGORITMA UNTUK MEMBANGUN POHON MERENTANG MINIMUM

Ada beberapa algoritma untuk mendapatkan pohon merentang minimum dari graf G. Algoritma prim, kruskal, boruvka, dan sollin dapat digunakan untuk mendapatkan pohon merentang minimum. Namun, algoritma yang sering digunakan adalah algoritma prim dan kruskal.

2.1 Algoritma Prim

Algoritma Prim merupakan algoritma untuk mencari jalur minimum dari graf G. Algoritma ini membentuk pohon merentang minimum langkah per langkah. Pada tiap langkah diambil sisi graf G yang memiliki bobot paling minimum namun terhubung dengan pohon merentang minimum T yang telah terbentuk.

Berikut adalah langkah-langkah algoritma prim

1. Ambil sisi graf G yang memiliki bobot paling minimum, masukan ke dalam T
2. Pilih sisi (*u,v*) yang memiliki bobot minimum dan bersisian dengan simpul di T, tetapi tidak membentuk sirkuit di T
3. Tambahkan (*u,v*) ke dalam T
4. Ulangi sebanyak n-2 kali.

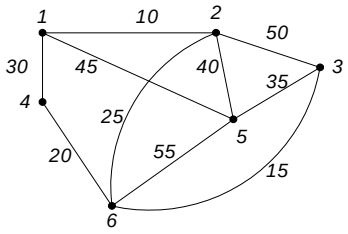
Algoritma prim dalam notasi *pseudo-code*-nya adalah sebagai berikut

```
procedure Prim(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |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 ← {(p,q)}
  for i←1 to n-2 do
    Pilih sisi (u,v) dari E yang bobotnya terkecil namun
    bersisian dengan simpul di T
    T ← T ∪ {(u,v)}
  endfor
```

Algoritma 1. Algoritma Prim untuk pohon merentang

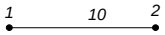
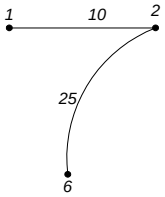
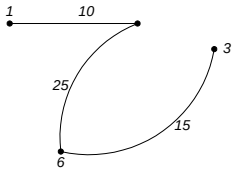
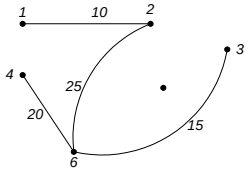
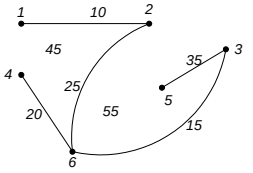
Adapun penerapan dari algoritma ini adalah berikut ini
Diberikan sebuag graf G



Gambar 2. Contoh Graf G

Penggunaan algoritma primakan diperoleh pohon merentang minimum dengan langkah berikut ini

Tabel 1. Pembentukan Pohon merentang dari graf G

Langkah	Sisi	Bobot	Pohon rentang
1	(1, 2)	10	
2	(2, 6)	25	
3	(3, 6)	15	
4	(4, 6)	20	
5	(3, 5)	35	

2.2 Algoritma Kruskal

Algoritma kruskal bekerja dengan mengurutkan sisi-sisi graf terlebih dahulu berdasarkan bobotnya dari yang terkecil sampai yang terbesar. Sisi-sisi yang dimasukkan ke dalam himpunan T adalah sisi-sisi graf sehingga T adalah pohon. Keadaan awalnya, sisi-sisi graf membentuk

hutan. Hutan tersebut merupakan hutan merentang. Sisi dari graf G yang dimasukkan ke T jika tidak membentuk siklus di T.

Langkah algoritma ini

1. T masih kosong
2. Pilih sisi (*u,v*) dengna bobot paling minimum yang tidak membentuk siklus di T dan tambahkan ke T
3. Ulangi sebanyak n-1 kali

Pseudo-code algoritma ini sebagai berikut

```
procedure Kruskal(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung -berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |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 - dari bobot kecil ke bobot
    besar)
  T ← {}
  while jumlah sisi 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
```

Algoritma 2. Algoritma Kruskal

Graf G (gambar 2) dapat diselesaikan dengan algortima kruskal berikut ini

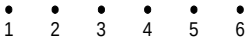
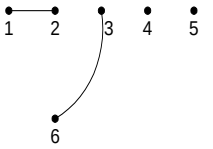
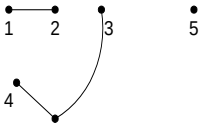
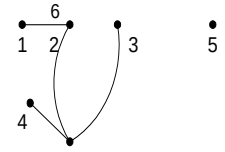
1. Hasil pengurutan sisi

Tabel 2. Pengurutan Sisi Graf

Sisi	(1, 2)	(3, 6)	(4, 6)	(2, 6)	(1, 4)	(3, 5)	(2, 5)	(1, 5)	(2, 3)	(5, 6)
Bobot	10	15	20	25	30	35	40	45	50	55

Dari hasil pengurutan akan diperoleh himpuan T yang merupakan pohon sehingga tidak ada sisi yang membentuk sirkuit.

Tabel 2. Pohon merentang dengan algoritma kruskal

Langkah	Sisi	Bobot	Hutan merentang
0			
1	(1, 2)	10	
2	(3, 6)	15	
3	(4, 6)	20	
4	(2, 6)	25	

2.3 Algoritma Boruvka

Algoritma ini mengecek bobot masing-masing sisi. Sisi yang minimum digabungkan dengan sisi yang minimum lainya yang keduanya memiliki hubungan *disjoint*. Algoritma ini memiliki *pseudo-code* sebagai berikut

1. Mulai dengan graf berbobot G, dan himpunan sisi T yang masih kosong
2. Sementara simpul dari graf G *disjoint* dengan T
    a. Mulai dengan himpunan sisi E yang masih kosong
    b. Untuk setiap komponen
        i. Mulai dengan himpunan sisi S yang masih kosong
        ii. Untuk setiap simpul dalam komponen tersebut
            1. Tambahkan sisi yang paling kecil bobotnya ke simpul yang lain yang *disjoint* dengan komponen S
        iii. Tambahkan sisi S ke E
    c. Tambahkan sisi S ke E
3. Tambahkan sisi E ke T
4. T merupakan pohon merentang minimum

Algoritma ini memiliki kompleksitas O(log V). V merupakan jumlah sisi pada graf.

2.4 Algoritma Sollin

Algoritma ini merupaka algoritma turunan dari algoritma boruvka. Ide algoritma ini yaitu menghapus sisi-sisi yang memiliki bobot paling besar dan tidak menyebabkan graf tidak terputus.

Langkah Algoritma ini adalah sebagai berikut

1. Ambil graf G
2. Tentukan sisi (*u*,*v*) yang memiliki bobot paling besar
3. Hapus sisi (*u*,*v*) jika tidak menyebabkan graf tidak terputus, dan jangan dihapus jika sebaliknya
4. Lakukan sebanyak n kali

Pseudo-code dari algoritma sollin ini adalah sebagai berikut

```
Procedure Sollin(input G : graf, output T : pohon)
{Masukan graf G-berbobot yang terhubung}
{Keluarannya merupakan pohon merentang minimum}
{MEmbentuk pohon merentang minimum T dari graf G}
Kamus local
U : integer;

Algoritma
T<-{G}
While (u<=n) do
Pilih sisi (u,v) terbesar bobotnya dari G
If (u,v) tidak meyebabkan tidak terhubung jika dihapus then
      T<- T-{(u,v)}
Endif
Endwhile
```

Algoritma 3. Algoritma Sollin untuk pohon merentang

Contoh aplikasi dari graf ini dengan menggunakan Graf G (gambar 2)

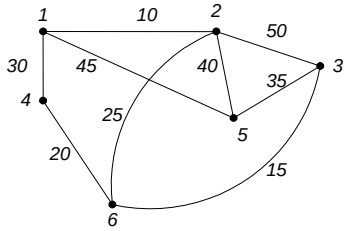
1. Mengurutkan Bobot sisi mulai dari yang terbesar

Tabel 3. Pengurutan sisi berdasarkan bobotnya

Bobot	Sisi
55	(5,6)
50	(2,3)
45	(1,5)
40	(2,5)
35	(3,5)
30	(1,4)
25	(2,6)
20	(4,6)
15	(3,6)
10	(1,2)

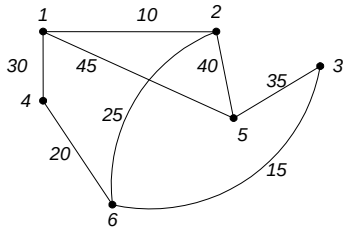
2. Menghapus sisi dengan bobot paling besar yang tidak menyebabkan graf tidak terhubung

   a. Sisi (5,6) yang akan dihapus pertama kali



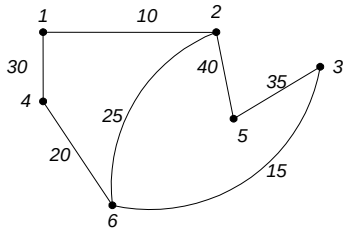
Gambar 3. Graf G sisi (5,6) telah dihapus

   b. Sisi (2,3) akan dihapus



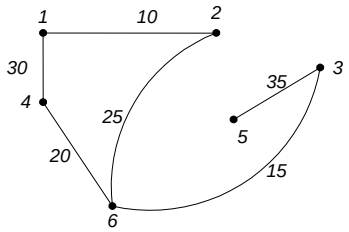
Gambar 4. Graf G sisi (2,3) telah dihapus

   c. Sisi (1,5) akan dihapus



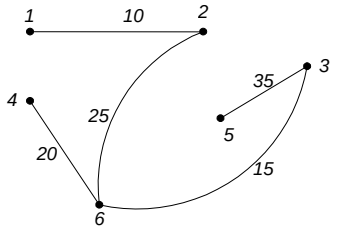
Gambar 5. Graf G sisi (1,5) telah dihapus

   d. Sisi (2,5) akan dihapus



Gambar 6. Graf G sisi (2,5) telah dihapus

e. Sisi (1,4) akan dihapus



Gambar 7. Graf G sisi (1,4) telah dihapus

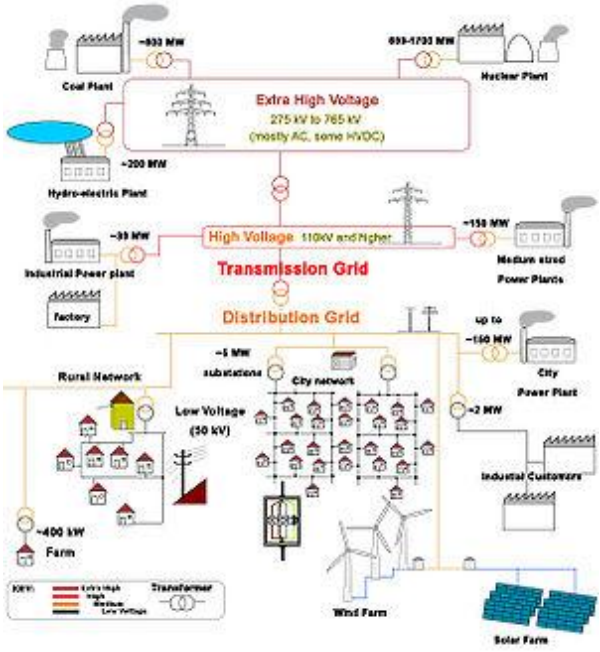
Sisi tersebut merupakan sisi yang terakhir dihapus. Hasilnya adalah pohon merentang T dari graf G. Hasilnya sesuai dengan algoritma sebelumnya.

3. JARINGAN DISTRIBUSI LISTRIK

Sistem jaringan distribusi listrik dapat diilustrasikan sebagai graf. Sisi-sisi menggambarkan hubungan antara gardu listrik dengan rumah-rumah pelanggan atau gardu listrik dengan gardu-gardu utama, atau gardu-gardu utama dengan sumber pembangkit listrik.

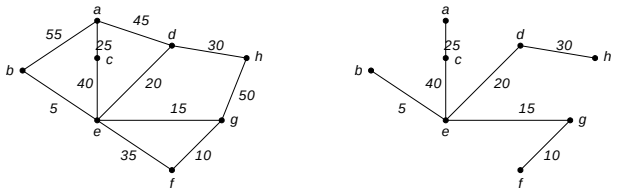
Pada pendistribusian listrik, listrik harus didistribusikan dari sumber pembangkit listrik ke pelangganya. Pada prosesnya pendistribusian ini tidak bisa langsung dari pembangkit listrik ke rumah-rumah melainkan melalui gardu-gardu listrik utama. Dari gardu listrik utama baru disampaikan ke gardu-gardu listrik dan pada akhirnya sampai ke rumah-rumah warga atau pabrik.

Untuk menghemat biaya dalam pendistribusian listrik dibutuhkan suatu jalur yang paling dekat dari semua jalur suatu rumah ke gardu listrik. Penyelesaian permasalah ini sangat cocok dengan pohon merentang minimum.



Gambar 8. Topologi Jaringan Distribusi Listrik

Gardu-gardu listrik, tiang listrik, dan pelanggan diibaratkan sebagai simpul sedangkan kawatnya diibaratkan sebagai sisinya. Semakin pendek jarak distribusinya, semakin murah biaya yang dikeluarkan.

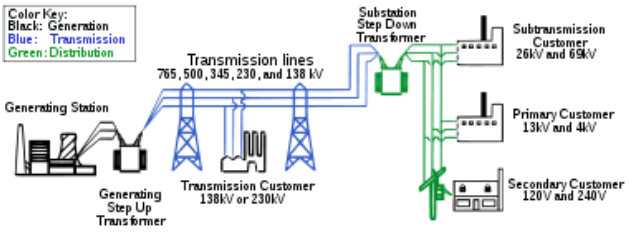


Gambar 9. Jaringan distribusi listrik digambarkan dalam graf

Contoh menggunakan graf di atas. Simpul e merupakan gardu listrik sedangkan simpul lainya merupakan pelanggan. Ada banyak kemungkinan jalur yang digunakan tetapi jalur yang paling efektif adalah jalur yang merupakan pohon merentang minimum.

Pada jaringan distribusi listrik akan ditemukan pohon merentang minimum yang sangat besar karena merupakan gabungan dari pohon merentang minum yang kecil. Sebagai contoh jalur distribusi listrik Jawa-Bali.

Besarnya tegangan masing-masing gardu atau tiang listrik diilustrasikan dengan gambar dibawah ini



Gambar 10. Besar tegangan pada gardu listrik

4. KESIMPULAN

Berdasarkan uraian di atas, pohon merentang minimum dapat dibangun dari graf berbobot dengan menggunakan algoritma prim, kruskal, boruvka, dan sollin atau gabungan dari algoritma tersebut. Pohon merentang minimum ini dapat dimanfaatkan untuk menyelesaikan persoalan jalur distribusi listrik dengan mencari jalur yang paling efektif sehingga biaya yang digunakan untuk membangun jalur tersebut lebih murah.

REFERENSI

[1] Munir, Rinaldi, "Matematika Diskrit", Informatika, 2003.
[2] en.wikipedia.org/electricity_network : 18 Desember 2009
[3] en.wikipedia.org/Borůvka's_algorithm : 18 Desember 2009
[4] dunia-listrik.blogspot.com/2008/ : 18 Desember 2009
[5] basementbrat.wordpress.com/2008 : 18 Desember 2009
[6] imadudd1n.wordpress.com/2008/03 : 18 Desember 2009
[7] pages.cs.wisc.edu/-elgar/cs557/Sollin/ : 18 Desember 2009