Permodelan Pohon Merentang Minimum Dengan Menggunakan Algoritma Prim dan Algoritma Kruskal

Salman Muhammad Ibadurrahman – NIM : 13506106

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-MAIL : if16106@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang pamecahan masalah graf dengan memodelkannya sebagai pohon merentang minimum dengan menggunakan algoritma Prim. Permodelan pohon merentang adalah salah satu permodelan graf yang digunakan untuk mencari nilai minimum dari laluan pada suatu graf. Persoalan dunia nyata yang dapat di-*representasikan* ke dalam bentuk graf dapat dicari penyelesaian yang efisien dengan *representasi* graf pohon merentang dengan algoritma Prim. Di dalam makalah ini diterangkan pula tentang perbedaan antara permodelan pohon merentang dengan algoritma Prim, algoritma Kruskal.

Kata kunci: Pohon merentang, graf, simpul, sisi, algoritma Prim, algoritma Kruskal, improper subset (himpunan bagian tidak sebenarnya), proper subset (himpunan bagian sebenarnya), pohon berbobot.

1. Pendahuluan

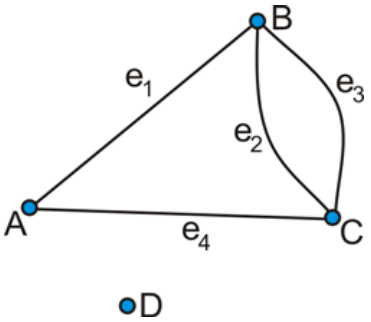
Pohon merentang adalah suatu pohon T yang berasal dari sebuah graf G. Graf G adalah graf tak terhubung yang bukan merupakan pohon (graf yang memiliki sirkuit). Sedangkan pohon T adalah graf yang merupakan graf pohon yang didapat dengan cara memutuskan sirkuit-sirkuit yang terdapat pada graf G. Jadi, pohon merentang adalah graf pohon yang himpunan semua simpulnya merupakan improper subset dari himpunan simpul yang terdapat pada graf G, sedangkan himpunan semua sisi pada pohon merentang merupakan proper subset dari himpunan semua sisi pada graf G.

2. Teori Graf

Graf (*Graph*) didefinisikan sebagai:

G = {V, E}

Dalam hal ini, V merupakan himpunan tidak kosong dari simpul-simpul (vertices / node) di gambarkan dalam titik-titik, dan E adalah himpunan sisi-sisi (edges / arcs) digambarkan dalam garis-garis yang menghubungkan sepasang simpul. Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi,



Gambar 1: Graf G1

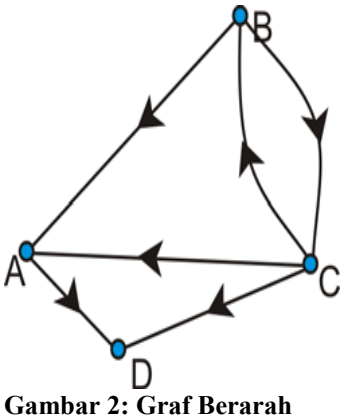
Pada G1 diatas, graf terdiri dari himpunan V dan E yaitu:

V = {A, B, C, D}
E = {e1, e2, e3, e4} ; bisa kita tulis
= {(A,B),(B,C),(B,C),(A,C)}

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:

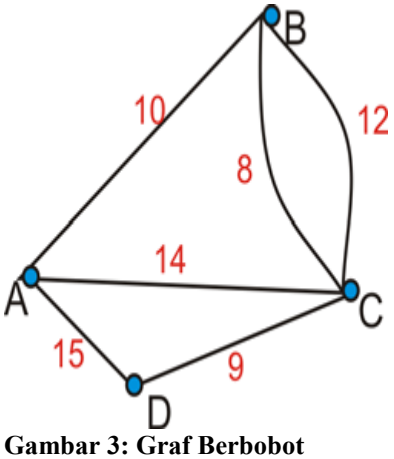
a. Graf Berarah (Directed Graph/Digraph)

Graf berarah adalah graf yang setiap sisinya diberi orientasi arah. Dalam hal ini sisi yang ditulis (v1,v2) berbeda dengan sisi (v2, v1)



Gambar 2: Graf Berarah

b. Graf Berbobot (Weight Graf)
Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga.



Gambar 3: Graf Berbobot

c. Graf Lengkap (Complete Graph)
Graf lengkap adalah graf sederhana, tidak mengandung gelang (sisi yang kedua simpulnya sama) maupun sisi ganda (dua sisi yang memiliki simpul asal dan simpul tujuan yang sama), serta setiap sisinya mempunyai sisi ke simpul lain. Gambar 4: Graf Lengkap K5 d. Bertetangga (Adjacent) Dua buah simpul pada graf tak berarah ikatakan bertentangga bila keduanya terhubung dengan sebuah sisi. Dapat dikatakan, jika ada v1 dan v2 yang bertetangga, maka harus ada sisi (v1,v2)

e. Bersisian (Incident)
Untuk sembarang sisi e = (v1,v2), sisi e dikatakan bersisian dengan simpul v1 dan simpul v2. f. Simpul Terpencil (Isolated Vertex) Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya. Dengan kata lain, simpul ini ialah simpul yang tidak satupun bertetangga dengan simpul-simpul lain.

f. Lintasan (Path)
Lintasan yang panjangnya n dari simpul awal v0 ke simpul akhir vn di dalam graf G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk v0, e1, v1, e2, …, vn-1, en, vn sedemikian sehingga e1=(v0,v1), e2=(v1,v2), …, n=(vn-1,vn) adalah sisi-sisi dari graf G.

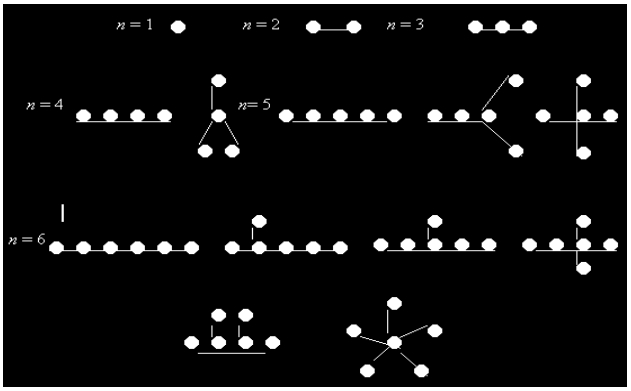
g. Siklus (Cycle) atau Sirkuit (Circuit)
Lintasan yang berawal dan berakhir pada simpul yang sama disebut siklus atau sirkuit.

h. Terhubung (Connected)
Graf disebut graf terhubung jika untuk setiap pasang simpul v1 dan v2 di dalam himpunan V terdapat lintasan dari v1 ke v2, yang juga berarrti ada lintasan dari v2 ke v1 (untuk graf berarah).

i. Upagraf (Subgraf) dan Komplemen Upagraf
Misalkan G = {V,E} sebuah graf, G1 = {V1,E1} dikatakan upagraf dari G jika V1 ⊂ v dan E1 ⊂ E. Komplemen dari upagraf G1terhadap G adalah G2 = {V2, E2}sedemikian sehingga E2 = E – E1dan V2 adalah himpunan simpul yang anggota-anggota E2 bersisian dengannya.

j. Upagraf Merentang (*spanning Subgraf*)
Upagraf G1 = {V1, E1} dari G = {V, E} disebut upagraf merentang jika, V1 = V, G1 mengandung semua simpul G.

k. Pohon (Tree)
Pohon adalah graf tak berarah terhubung yang tidak mempunyai sirkuit.



Gambar 4: Pohon

3. Pohon Merentang Minimum (PPM)

Jika G adalah graf berbobot, maka bobot dari pohon merentang T dari G didefinisikan sebagai jumlah bobot pada semua sisi di T. Pohon merentang yang berbeda memiliki bobot yang berbeda pula. Di antara pohon merentang di G, pohon yang memiliki bobot paling minimum dinamakan *pohon merentang minimum.* Pohon ini merupakan pohon merentang yang paling penting. Pohon merentang minimum memiliki terapan yang sangat luas dalam dunia nyata. Misalnya, pembangunan rel kereta api yang menghubungkan sejumlah kota. Pembangunan jalur kreta ini tidak perlu menghubingkan langsung dua buah kota; tetapi cukup membangun jalur kereta seperti pohon merentang. Karena di dalam graf mungkin terdapat beberapa pohon merentang, maka harus dicari pohon merentang minimum.

Terdapat dua buah algoritma untuk membangun pohon merentang minimum. Yaitu algoritma Prim dan algoritma Kruskal.

a. Algoritma Prim

Algoritma Prim membentuk pohon merentang minimum dengan langkah per langkah. Pada setiap langkah kita mengambil sisi graf G yang memiliki bobot minimum namun yang terhubung dengan pohin merentang T yang telah terbentuk.

Algoritma Prim

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

Jumlah seluruh langkah pada algoritma Prim ini adalah 1+(n-2) = n-1, yaitu sama dengan jumlah sisi pada pohon merentang dengan n buah simpul.

Algoritma Prim dalam *Pseudocode*:

```
procedure Prim (input G: graf,
output T: pohon)
{Membentuk pohon merentang
minimum T dari graf terhubung
G}
```

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

Algoritma:
Cari sisi (p, q) dari E yang
memiliki bobot terkecil
T <- {(p, q)}
for
     Pilih sisi (u,v) yang
     memiliki bobot paling
     kecil dan bersisian dengan
     simpul di T
     T <- T U {(u, v)}
endfor
```

.b Algoritma Kruskal (Kruskal's Algorithm)

Algoritma Kruskal adalah juga tergolong algoritma greedy dalam teori graf yang digunakan untuk mencari pohon merentang minimum. Algoritma ini pertama kali muncul pada tahun 1956 dalam sebuah tulisan yang ditulis oleh Joseph Kruskal.

Algoritmanya:

1. Himpunan sisi dari G diurutkan membesar sesuai bobot sisi tersebut.
2. Buat T dengan memasukkan 1 sisi terpendek dari G tersebut.
3. Ulang (banyak sisi T = (banyak simpul G) -1)

a. Ambil sisi selanjutnya dari G.
b. Jika sisi itu tidak membuat sirkuit di T
   - Masukkan sisi itu ke T.
   - Masukkan simpul-simpul sisi itu ke T.

Algoritma Kruskal dalam *pseudocode*:

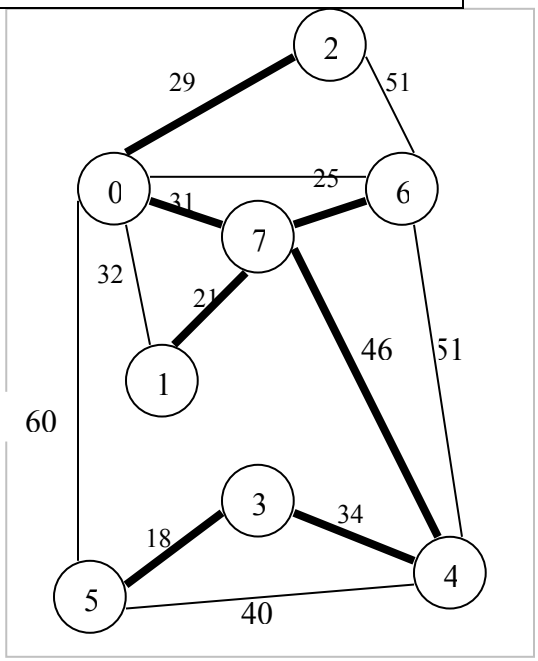
```
procedure Kruskal (input G: graf, output T: pohon)
{Membentuk pohon merentang minimum T dari graf terhubung G }

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

Algoritma:
{Asumsi: sisi-sisi graf sudah diurut menaik berdasarkan bobotnya}

T <- {}
while jumlah sisi T < n-1 do
     Pilih sisi dari E yang bobotnya terkecil

     if (u, v) tidak membentuk siklus di T then
        T <- T U {(u, v)}
     endif
endwhile
```

4. Perbandingan 2 Algoritma

Studi Kasus

Saat teknologi komunikasi mulai berkembang, pemerintah ingin mengganti sistem jaringan kabel telepon yang sudah ada di satu daerah dengan kabel yang baru. Namun, karena ini baru merupakan awal, pemerintah tidak mau ambil resiko mengganti semua kabel yang menghubungkan wilayah satu dengan yang lain. Ia hanya akan memasang jaringan di wilayah itu jika wilayah itu memang belum tersentuh jaringan yang baru.

Bantulah pemerintah untuk membangun jaringan yang menghubungkan semua wilayah dari jaringan lama yang diberikan. Dengan asumsi, semakin panjang kabel yang dipasang, semakin mahal biaya yang harus dikeluarkan, buatlah jaringan yang dibangun ini memakan biaya sesedikit mungkin.



Gambar 5: Graf sistem jaringan kabel telepon

Penyelesaian dengan algoritma Prim

langkah	e	V-e	Bobot
1	{0}	{1,2,3,4,5,6,7}	29
2	{0,2}	{1,3,4,5,6,7}	31
3	{0,2,7}	{1,3,4,5,6}	21
4	{0,1,2,7}	{3,4,5,6}	25
5	{0,1,2,6,7}	{3,4,5}	46
6	{0,1,2,4,6,7}	{3,5}	34
7	{0,1,2,3,4,6,7}	{5}	18
8	{0,1,2,3,4,5,6,7}	{}	204

Tabel 1: Penyelesaian dengan algoritma Prim

Penyelesaian dengan algoritma Kruskal

langkah	sisi minimum	bobot	hasil sisi pada T
0			{}
1	{5,3}	18	{(5,3)}
2	{7,1}	21	{(5,3),(7,1)}
3	{7,6}	25	{(5,3),(7,1),(7,6)}
4	{0,2}	29	{(5,3),(7,1),(7,6),(0,2)}
5	{7,0}	31	{(5,3),(7,1),(7,6),(0,2),(7,0)}
6	{4,3}	34	{(5,3),(7,1),(7,6),(0,2),(7,0),(4,3)}
7	{7,4}	46	{(5,3),(7,1),(7,6),(0,2),(7,0),(4,3),(7,4)}
Jumlah bobot		204	

Tabel 2: Penyelesaian dengan algoritma Kruskal

Kesimpulan

Antara algoritma Prim dan algoritma Kruskal memiliki perbedaan, yaitu, langkah-langkah yang diambil oleh masing-masing algoritma, jumlah langkah, dan cara penentuan pohon merentang minimum.

Algoritma Prim lebih berorientasi pada pencarian simpul-simpul yang dihubungkan oleh sisi dengan bobot minimum. Sedangkan pada algoritma Kruskal lebih berorientasi pada pencarian sisi-sisi yang memiliki bobot minimum lalu mengurutkannya. Pada algoritma Kruskal, proses mengurutkan inilah yang membuat ini kalah efisien dibandingkan dengan algoritma Prim. Ini terlihat dari jumlah langkah dari masing-masing algoritma. Algoritma Prim membutuhkan (n-1) langkah, sedangkan algoritma Kruskal membutuhkan n langkah.

Daftar Pustaka

[1] Munir, Rinaldi.(2005).Bahan Kuliah IF2151 Matematika Diskri. Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[2] English Wikipedia, Ensiklopedia Bebas *http://www.en.wikipedia.org/wiki/* Tanggal akses : 29 Desember 2007 pukul 16:30

[3] Kumpulan Makalah Matematika Diskrit *http://www.informatika.org/~rinaldi/matdis/* Tanggal akses: 29 Desember 2007 pukul 16:30