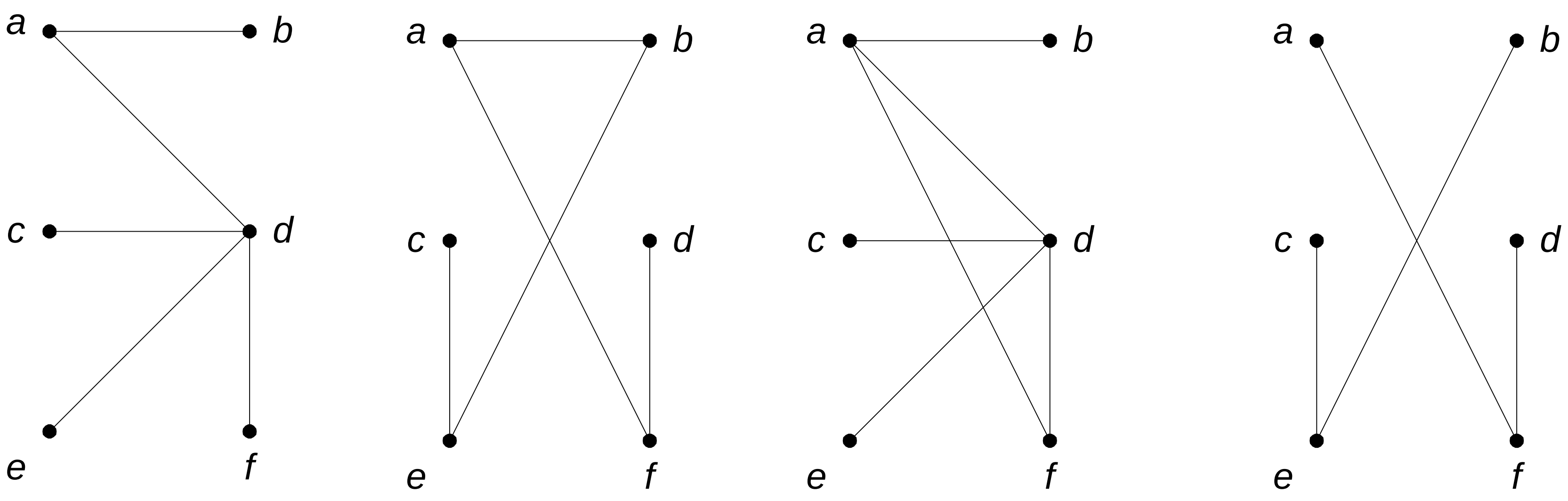
Pohon (Bag. 1)

Bahan Kuliah *IF2120 Matematika Diskrit*

Program Studi Teknik Informatika
STEI- ITB

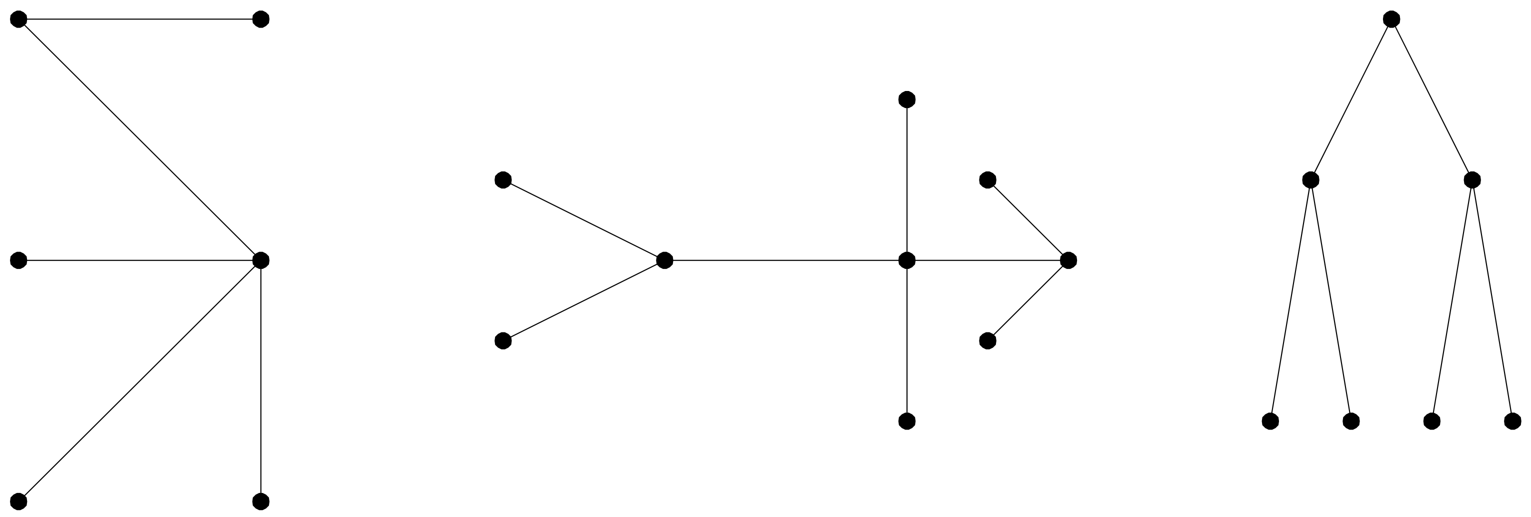
Definisi Pohon

- **Pohon** adalah graf tak-berarah terhubung yang tidak mengandung sirkuit



pohon pohon bukan pohon bukan pohon

Hutan (*forest*) adalah
- kumpulan pohon yang saling lepas, atau
- graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon.



Hutan yang terdiri dari tiga buah pohon

Hutan

Sifat-sifat (properti) pohon

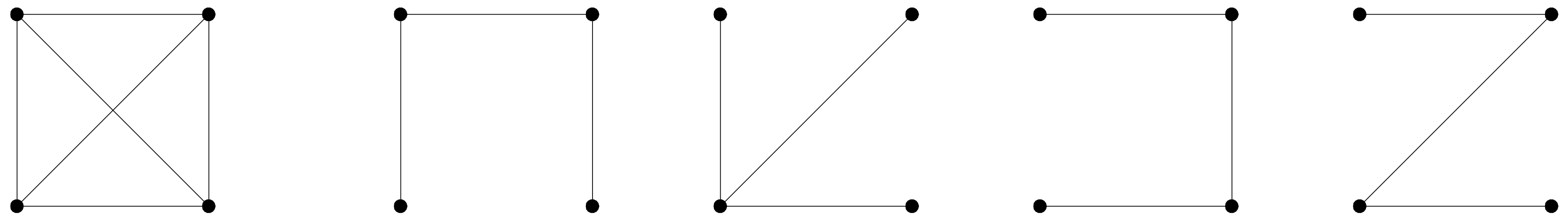
Teorema. Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n. Maka, semua pernyataan di bawah ini adalah ekivalen:

1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n – 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki $m = n – 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

- Teorema di atas dapat dikatakan sebagai definisi lain dari pohon.

Pohon Merentang (*spanning tree*)

- Pohon merentang dari graf terhubung adalah upagraf merentang yang berupa pohon.
- Pohon merentang diperoleh dengan memutus sirkuit di dalam graf.



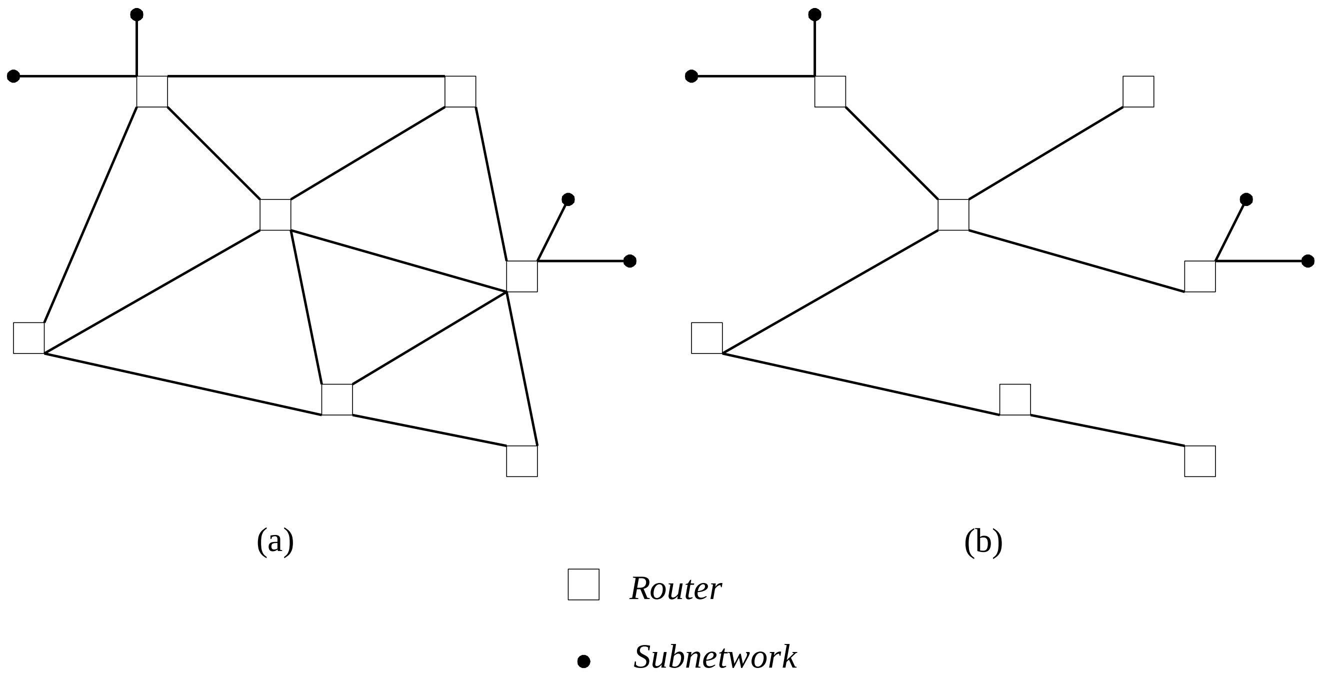
G T_1 T_2 T_3 T_4

- Setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang.

- Graf tak-terhubung dengan *k* komponen mempunyai *k* buah hutan merentang yang disebut hutan merentang (*spanning forest*).

Aplikasi Pohon Merentang

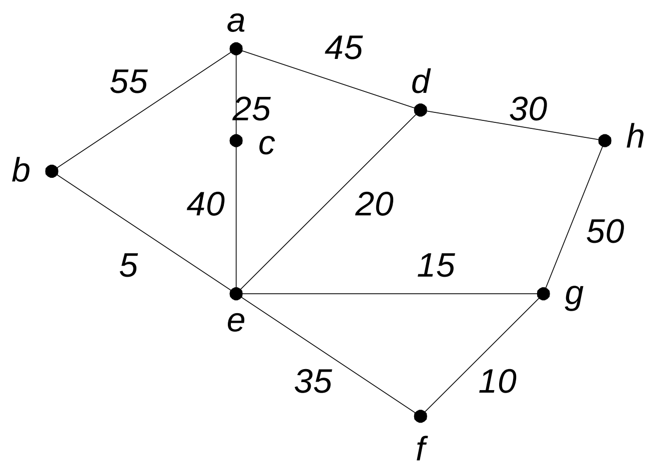
1. Jumlah ruas jalan seminimum mungkin yang menghubungkan semua kota sehingga setiap kota tetap terhubung satu sama lain.
2. Perutean (*routing*) pesan pada jaringan komputer.

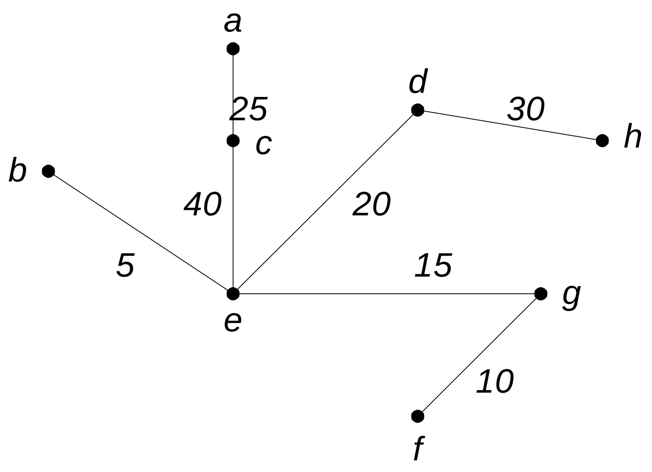


(a) Jaringan komputer, (b) Pohon merentang *multicast*

Pohon Merentang Minimum

- Graf terhubung-berbobot mungkin mempunyai lebih dari 1 pohon merentang.
- Pohon merentang yang berbobot minimum –dinamakan **pohon merentang minimum** (*minimum spanning tree*).





Algoritma Prim

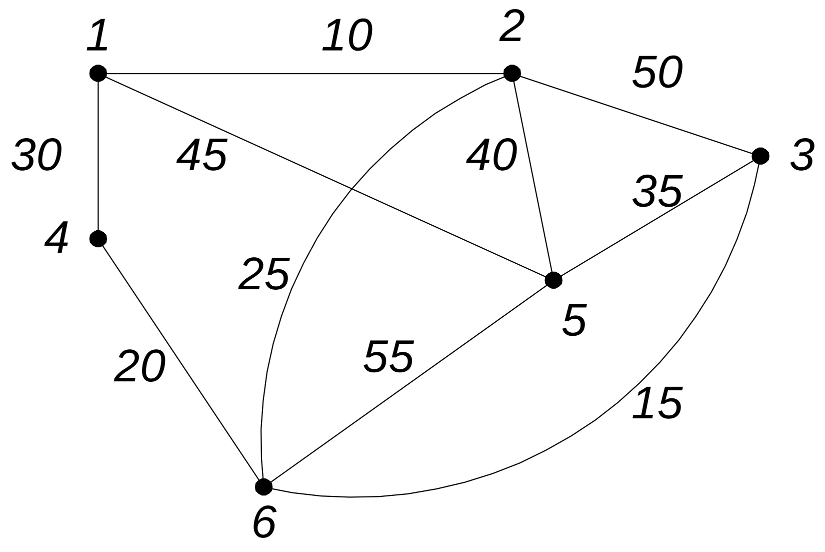
Langkah 1: ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T.

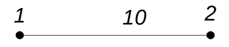
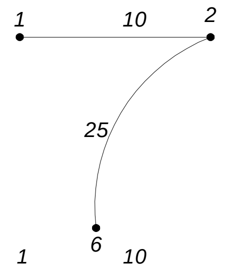
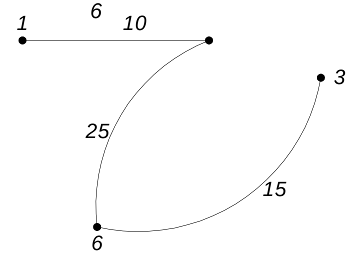
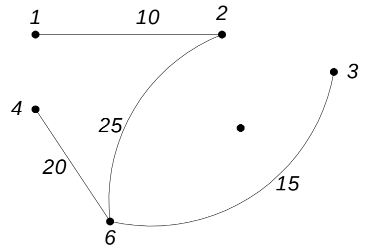
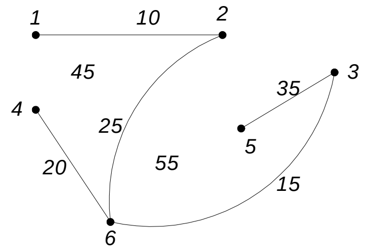
Langkah 2: pilih sisi (u, v) yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi (u, v) tidak membentuk sirkuit di T. Masukkan (u, v) ke dalam T.

Langkah 3: ulangi langkah 2 sebanyak $n – 2$ kali.

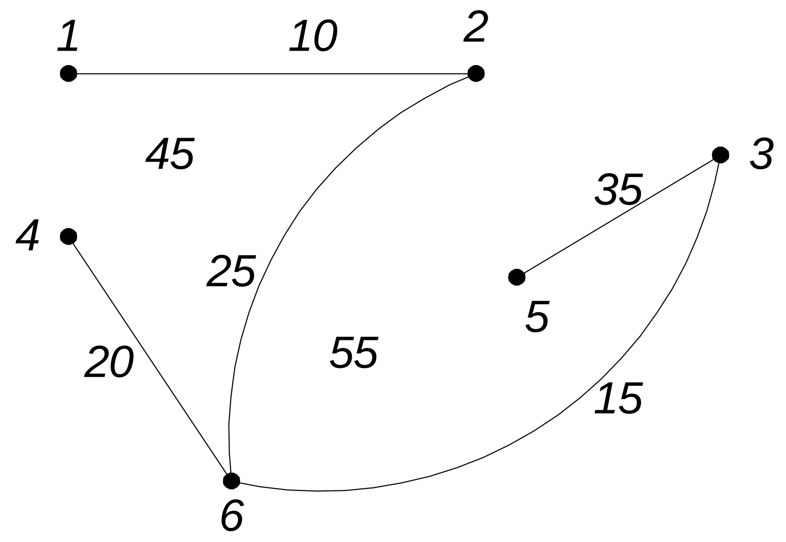
```
procedure Prim(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-
berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan /V/= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
  i, p, q, u, v : integer

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
```

Contoh:



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	

Pohon merentang minimum yang dihasilkan:

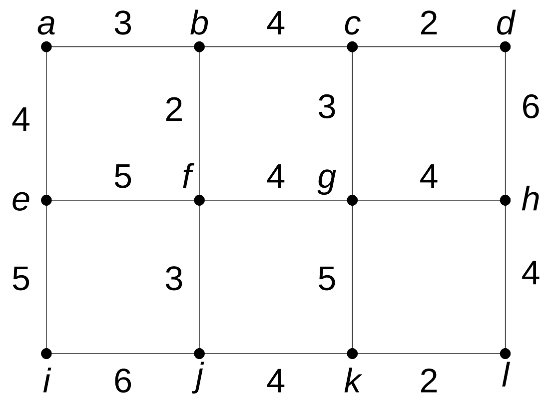


Bobot = 10 + 25 + 15 + 20 + 35 = 105

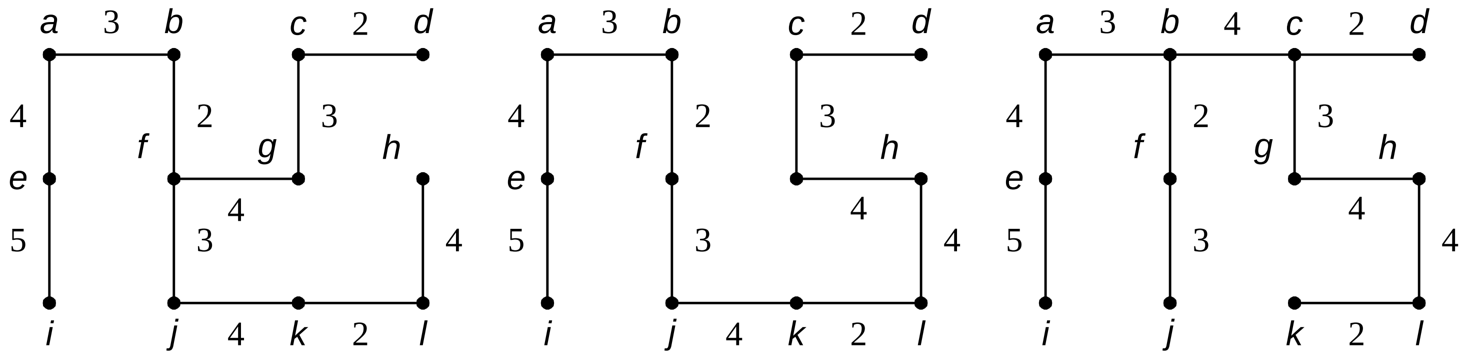
- Pohon merentang yang dihasilkan tidak selalu unik meskipun bobotnya tetap sama.

- Hal ini terjadi jika ada beberapa sisi yang akan dipilih berbobot sama.

Contoh:



Tiga buah pohon merentang minimumnya:



Bobotnya sama yaitu = 36

Algoritma Kruskal

(Langkah 0: sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya – dari bobot kecil ke bobot besar)

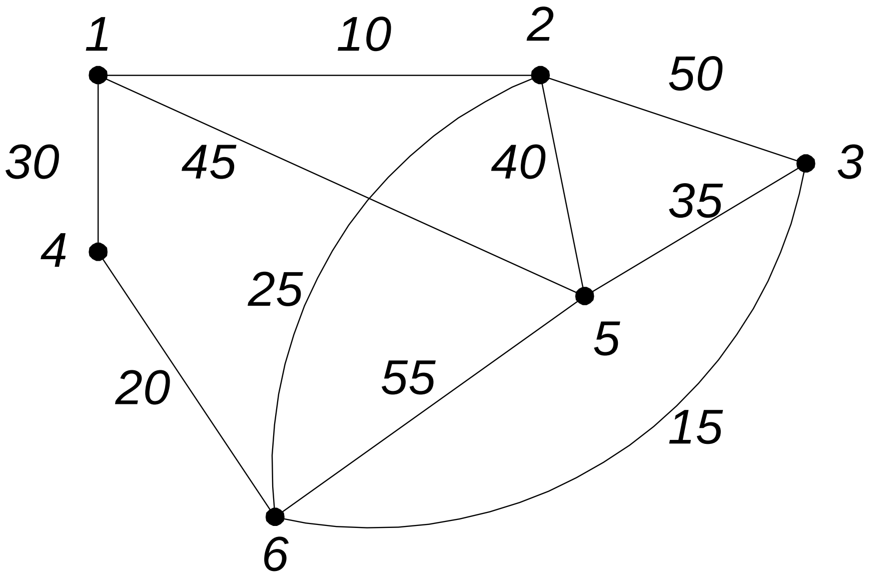
Langkah 1: T masih kosong

Langkah 2: pilih sisi (u, v) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (u, v) ke dalam T.

Langkah 3: ulangi langkah 2 sebanyak n – 1 kali.

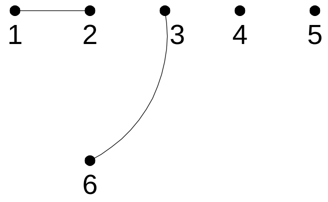
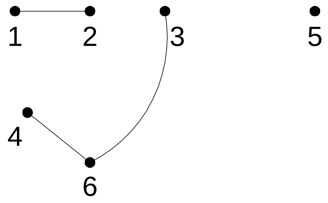
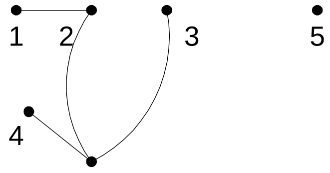
```
procedure Kruskal(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung –
berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
  i, p, q, u, v : integer

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
```

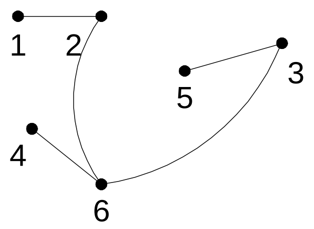
Contoh:



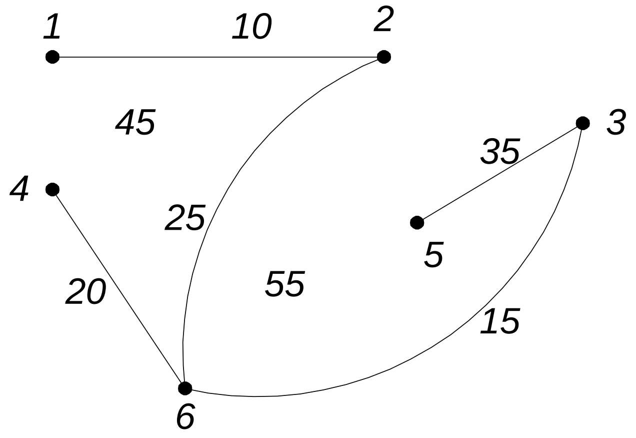
Sisi-sisi diurut menaik:

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

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

5	(1, 4)	30	ditolak
6	(3, 5)	35	

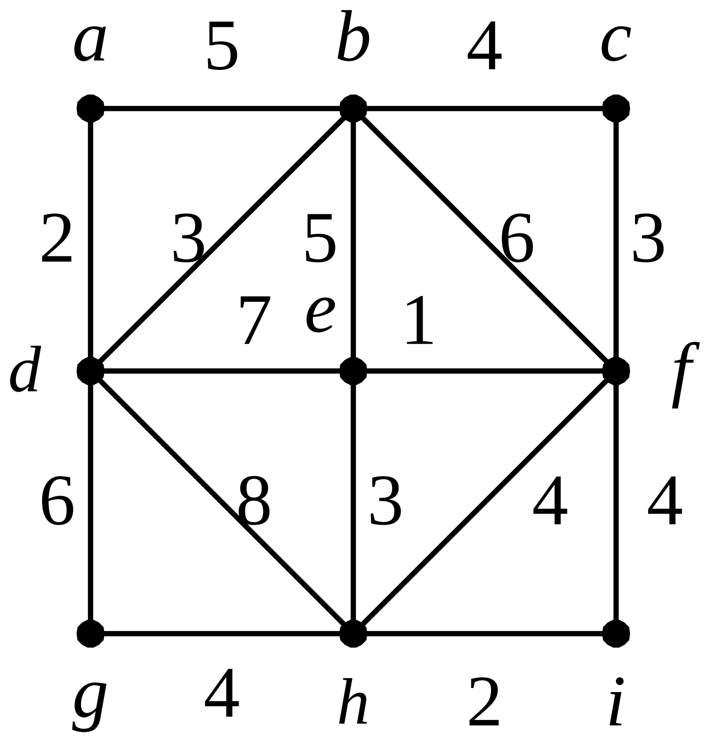
Pohon merentang minimum yang dihasilkan:



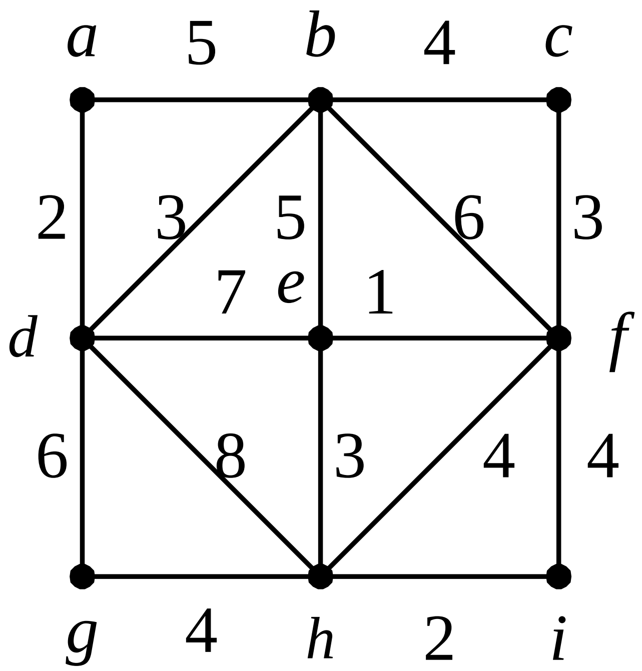
Bobot = 10 + 25 + 15 + 20 + 35 = 105

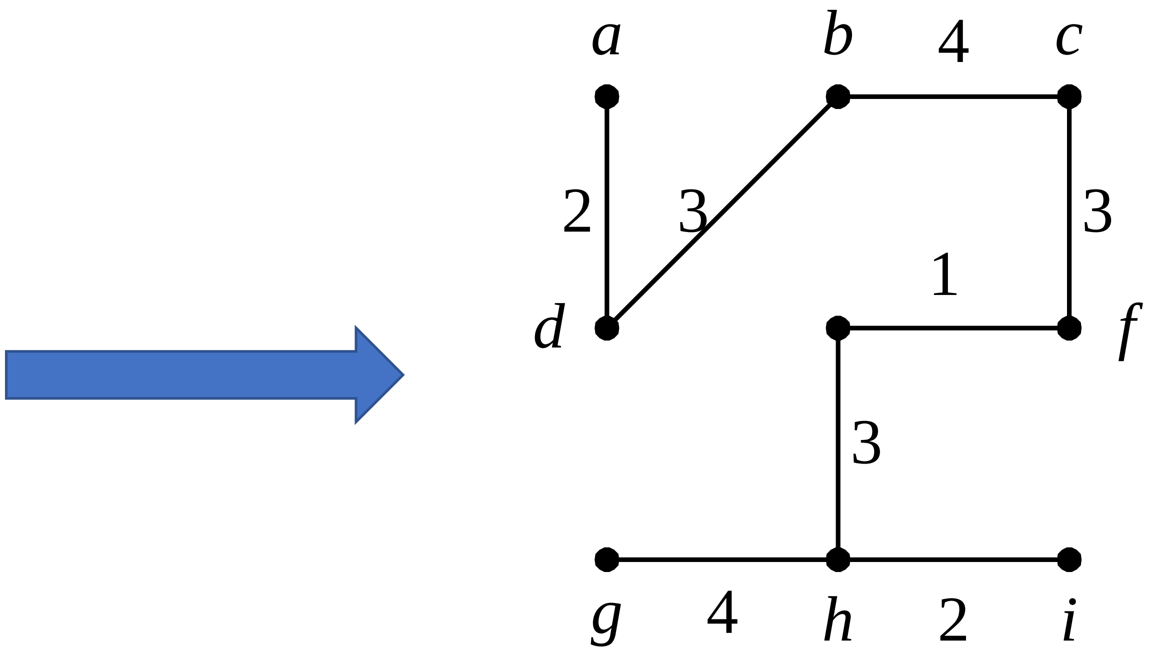
Latihan

Tentukan dan gambarkan pohon merentang minimum dari graf di bawah ini (tahapan pembentukannya tidak perlu ditulis).



Jawaban:





Bobot pohon merentang minimum: 1 + 3 + 3 + 2 + 4 + 4 + 3 + 2 = 22