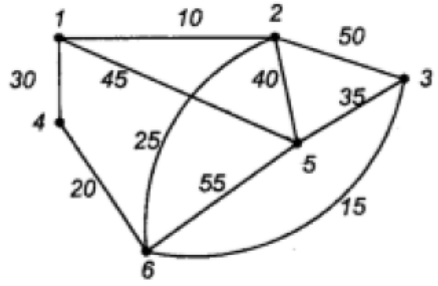
1. **Studi Kasus 1:**

Cari *minimum spanning tree* pada graf di bawah dengan Algoritma Kruskal. Jelaskan langkah demi langkah sampai graf membentuk *minimum spanning tree*.

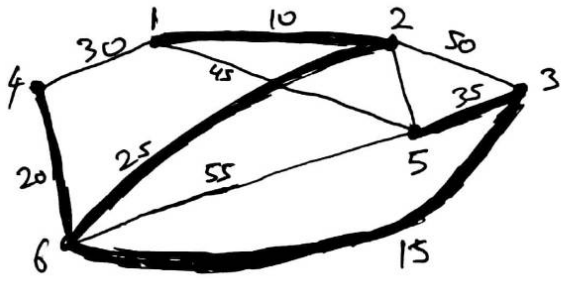


Langkah 1: Menghapus semua loop dan ujung parallel.

Langkah 2: Mengatur semua ujung dari yang terkecil hingga terbesar.

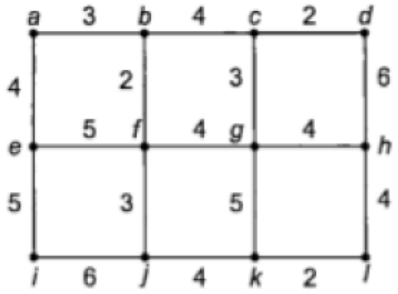
- 1, 2 = 10
- 3, 6 = 15
- 4, 6 = 20
- 2, 6 = 25
- 1, 4 = 30
- 3, 5 = 35
- 2, 5 = 40
- 1, 5 = 45
- 4, 5 = 50
- 5, 6 = 55

Langkah 3: Menambahkan ujung dengan bobot yang paling kecil sebanyak mungkin tetapi tidak boleh mengandung Cycle.



2. Studi Kasus 2:

Gambarkan 3 buah *minimum spanning tree* yang berbeda beserta bobotnya untuk graf di bawah dengan Algoritma Prim. Jelaskan setiap langkah untuk membangun *minimum spanning tree*.

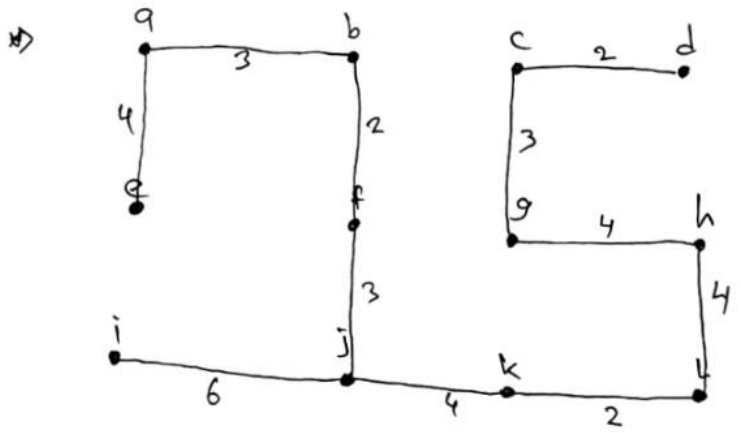


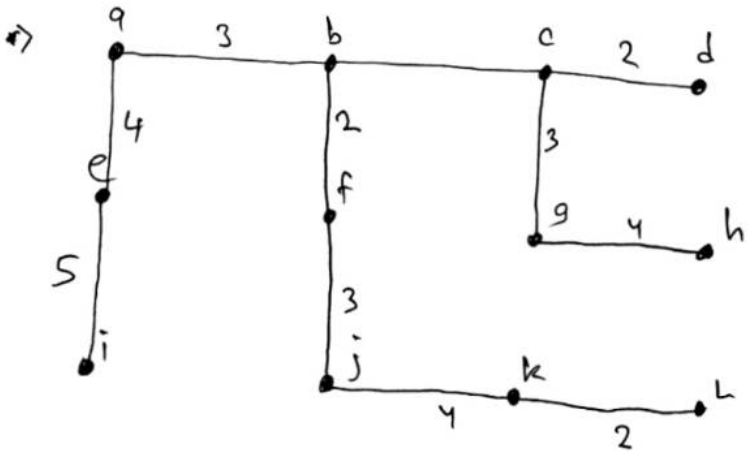
Langkah 1: Menentukan awal (Root Misalnya A).

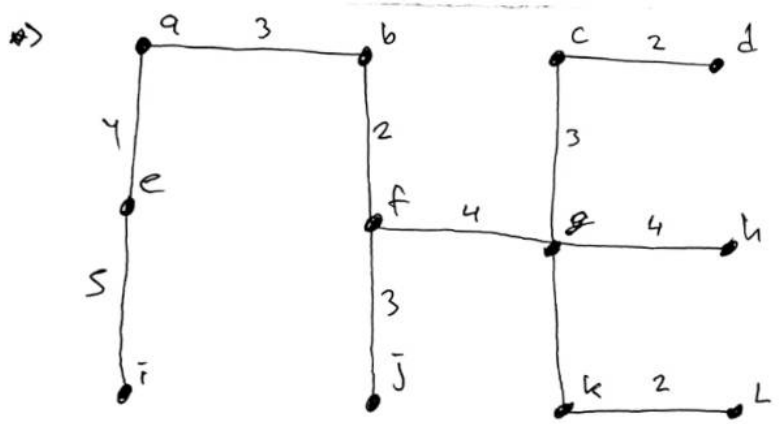
Langkah 2: Membuat Tree dengan satu ujung, temukan ujung dengan bobot paling kecil untuk menghubungkan simpul yang belum ada di tree.

Langkah 3: Ulangi langkah kedua sampai semua simpul terhubung.

Ada tiga kemungkinan hasil minimum spanning tree:







3. Studi Kasus 3:

Apakah semua *minimum spanning tree T* dari graf terhubung G harus mengandung jumlah sisi yang sama? Jelaskan alasannya (bukan dengan contoh).

Ya, karena syarat dari minimum spanning tree adalah setiap titik / simpul harus terhubung dan tidak boleh mengandung cycle. Maka jumlah sisi dari setiap minimum spanning tree pasti akan sama.