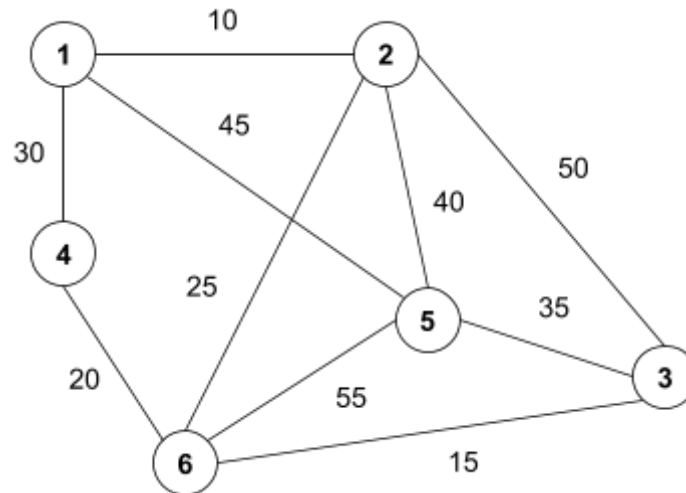


1. Kruskal's Algorithm

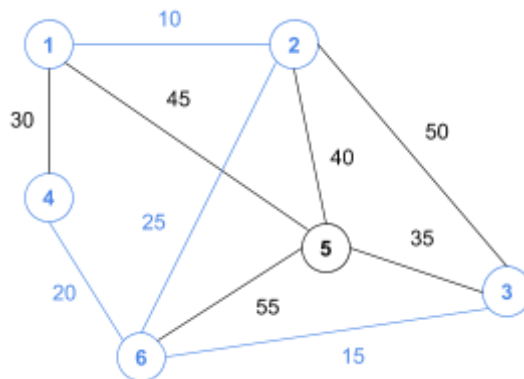
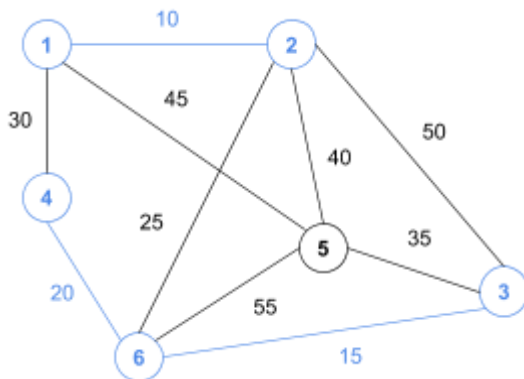
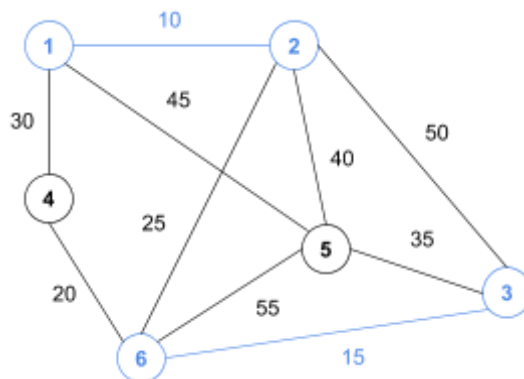
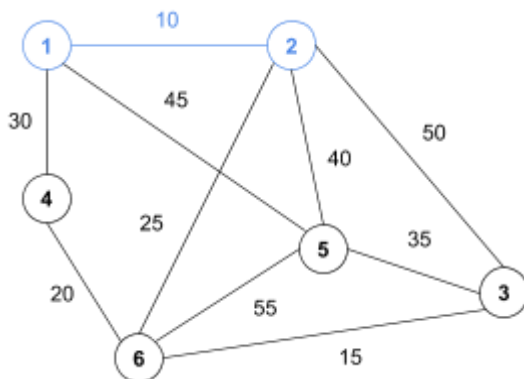


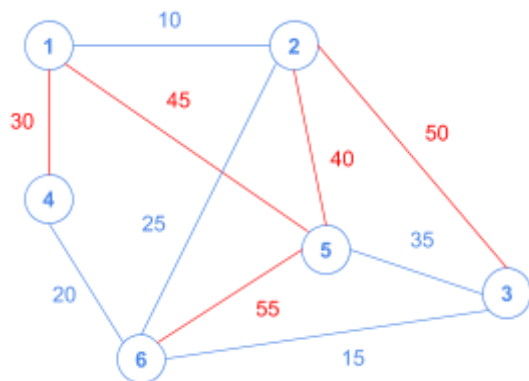
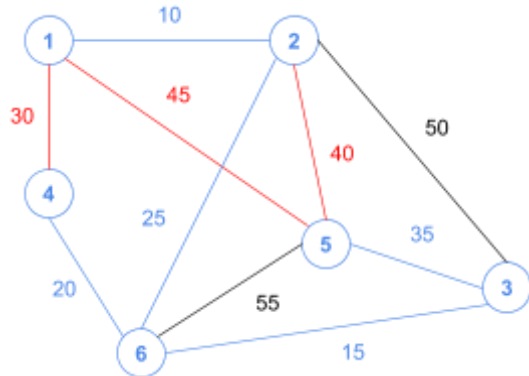
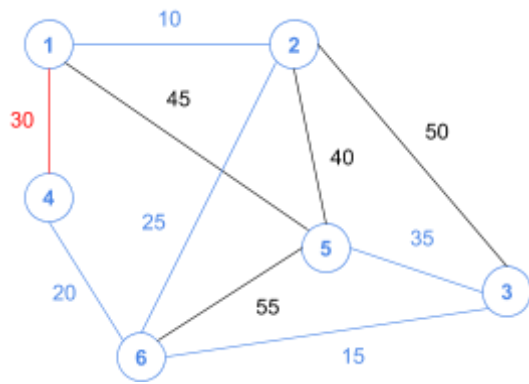
Urutkan tepi-tepi dari tepi terpendek.

(10,(0,1)), (15,(3,4)), (20,(2,3)), (25,(3,1)), (30,(0,2)),
(35,(5,4)), (40,(1,5)), (45,(0,5)), (50,(1,4)), (55,(5,6)).

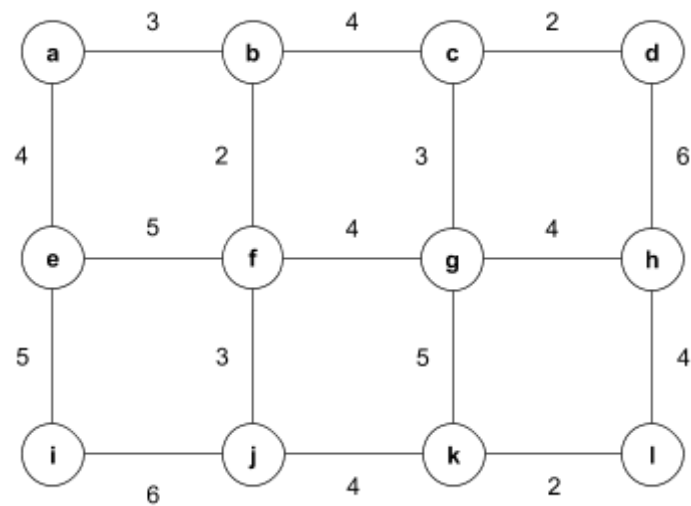
Cek setiap tepi sesuai urutan, masukkan ke dalam array T.

Jika tepi yang dicek membuat cycle dengan tepi sebelumnya, lewati.



[illegible]

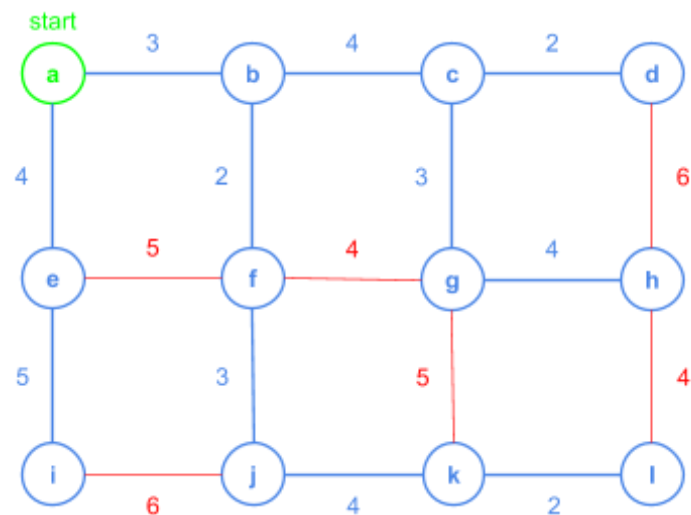
2. Prim's Algorithm



Langkah langkah :

- Menentukan node s, node awal.
- Tambahkan node yang tersambung ke node s ke PQ, diurutkan dari node terpendek.
- Cek node. Jika belum tersambung dalam tree T, Maka sambungkan dan hapus node dari PQ lalu tambahkan node yang tersambung ke node yang dihapus ke PQ. Jika sudah tersambung, lewati.

MST 1 : s = a; Panjang Tree = 36



MST 2 : s = d; Panjang Tree = 36

