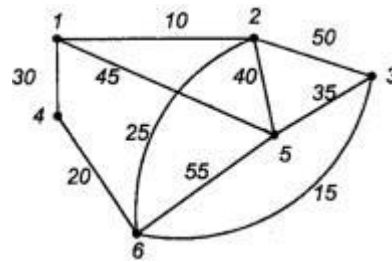


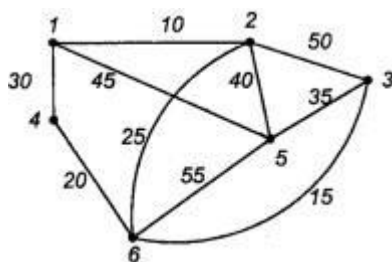
Nama : Irfan Satrio N
 NPM : 140810180003
 Kelas : A

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



Jawab :

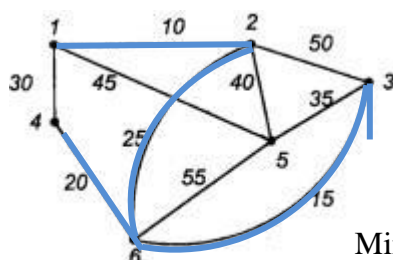
Langkah 1 – Menghapus semua *loop* dan *parallel edges*.



Langkah 2 – Mengatur semua *edge* pada graf dari yang terkecil ke 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
2,3	50
5,6	55

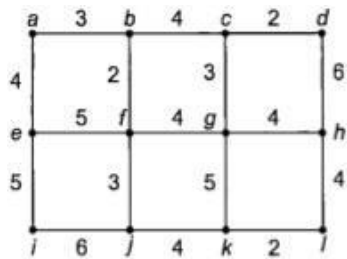
Langkah 3 – Menambahkan *edge* dengan bobot paling kecil. , lakukan dan jangan sampai membentuk sirkuit



Minimum spanning tree terbentuk

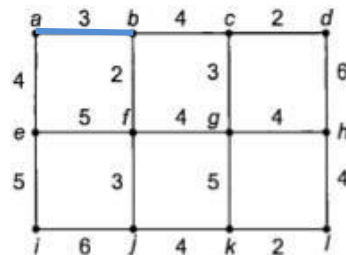
Nama : Irfan Satrio N
 NPM : 140810180003
 Kelas : A

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*.

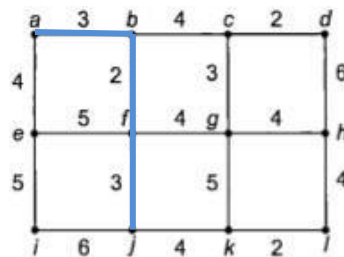


Jawab :

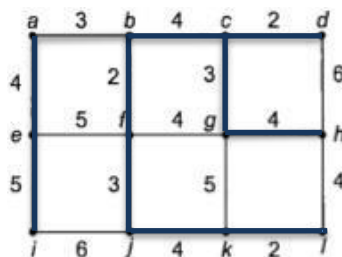
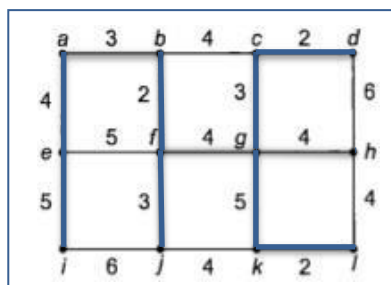
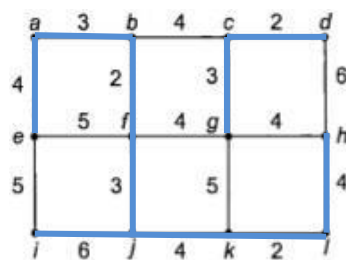
- a. Menentukan titik awal graph



- b. Menentukan simpul dengan minimum key value



- c. Ulangi langkah kedua sampai semua terdapat tree



Nama : Irfan Satrio N
NPM : 140810180003
Kelas : A

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

Jawab :

Setiap minimum spanning Tree T yang terbentuk dari graf terhubung G pasti mengandung jumlah sisi yang sama karena setiap minimum Tree memiliki aturan:

$$|E| = |V| - 1$$

dengan: $|E|$ = jumlah edges / sisi
 $|V|$ = jumlah vertex / node

Atau secara intuisi kita menyebutkan bahwa setiap vertex/node pada minimum spanning tree pasti terhubung dengan maksimal sebuah vertex/node lainnya, sehingga jumlah edges / sisi pasti sejumlah vertex/node di kurang satu

Nama : Irfan Satrio N
NPM : 140810180003
Kelas : A