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

1. **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 :**

1. Menentukan titik awal graph



1. Menetukan simpul dengan minimum key value



1. Ulangi langkah kedua sampai semua terdapat tree





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

**Jawab :**

Iya mengandung jumlah yang sama, karena dalam algoritma tersebut memiliki tujuan untuk mengunjungi semua titik dengan beban yang minimum sehingga semua titik dalam graph tersebut dapat dikunjungi