

METODE GREEDY

Metode ini digunakan untuk memperoleh solusi yang optimal dari suatu masalah yang mempunyai 2 indikator yaitu adanya fungsi tujuan & pembatas (Constrain).

```

PROCEDURE GREEDY (A,n)
Solusi ← 0 (solusi awal)
FOR I ← 1 TO n DO
  X ← SELECT(A)
  IF FEASIBLE (Solusi, x)
    THEN Solusi ← UNION (solusi, x)
  ENDIF
REPEAT
RETURN (Solusi)
END GREEDY
  
```

Keterangan :

A(1:n) mengandung n input data.

FEASIBLE merupakan fungsi yang bernilai boolean (0 atau 1)

UNION penggabungan dan pemeriksaan fungsi obyektifnya (update)

SELECT merupakan fungsi untuk mengambil data input dari A

CONTOH :

Himpunan A merupakan himpunan pasangan terurut (x,y), yaitu { (2,1),(3,2),(7,1), dan (1,0)}. Dari data-data tersebut akan ditentukan suatu pasangan terurut yang memiliki jumlah x dan y yang minimum. Adapun batasan dari x dan y masing-masing lebih besar dari nol.

Penyelesaiannya :

```

Solusi ← 0
N = 1 : x=2 > 0 }
        Y=1 > 0 } FEASIBLE (solusi, x)
        Solusi ← {(2,1)}
  
```

```

N = 2 : x=3 > 0 }
        Y=2 > 0 } FEASIBLE (solusi, x)
        Solusi ← {(2,1),{3,2}}
  
```

```

N = 3 : x=7 > 0 }
        Y = 1 > 0 } FEASIBLE (solusi, x)
        Solusi ← {{2,1),(3,2),(7,1)}
  
```

```

N = 4 : x = 1 > 0 }
        Y = 0 > 0 } TIDAK FEASIBLE
        Solusi ← {(2,1),{3,2),(7,1)}
  
```

Dari himpunan solusi yang mungkin tersebut diperoleh solusi yang optimal (dalam hal ini minimum) adalah (2,1) yang jumlahnya sebesar $2 + 1 = 3$.

Jadi solusi = (2,1)

METODE GREEDY banyak digunakan dalam berbagai penyelesaian masalah, antara lain adalah :

1. Optimal Storage on Tapes Problem
2. Knapsack Problem
3. Minimum Spanning Tree Problem
4. Shortest Path Problem

Dalam hal ini hanya akan dibahas mengenai minimum Spanning Tree saja.

MINIMUM SPANNING TREE

Permasalahan umum dari minimum spanning tree adalah mencari minimum biaya (cost) spanning tree dari setiap ruas (edge) suatu graph yang membentuk pohon (tree).

Dalam mendapatkan solusi yang diharapkan maka akan dipilih ruas menurut kriteria optimisasi yang menghasilkan biaya minimum. Dengan demikian penambahan jumlah biayanya relatif kecil dari setiap ruas yang telah terpilih dan membentuk spanning tree.

Untuk masalah minimum spanning tree, syarat graph dapat dicari minimum spanning tree adalah :

- ◆ Graph harus terhubung
- ◆ Ruasnya punya bobot / nilai
- ◆ Graphnya tidak berarah

Algoritma yang dapat dipakai untuk menentukan minimum spanning tree adalah :

- ✓ algoritma Solin
- ✓ Algoritma Kruskal
- ✓ Algoritma Prim's

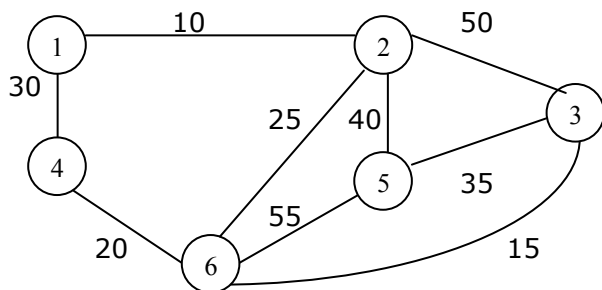
Dalam hal ini kita hanya membahas mengenai algoritma kruskal saja.

ALGORITMA KRUSKAL

Untuk mencari pohon rentang minimum dari graph dengan algoritma yang ditemukan Kruskal, mula-mula semua garis dalam graph diurut berdasarkan bobotnya dari kecil ke besar. Kemudian pilih garis dengan bobot terkecil. Pada setiap langkah dipilih garis dengan bobot terkecil, tetapi tidak membentuk loop garis-garis yang sudah dipilih terdahulu.

Contoh :

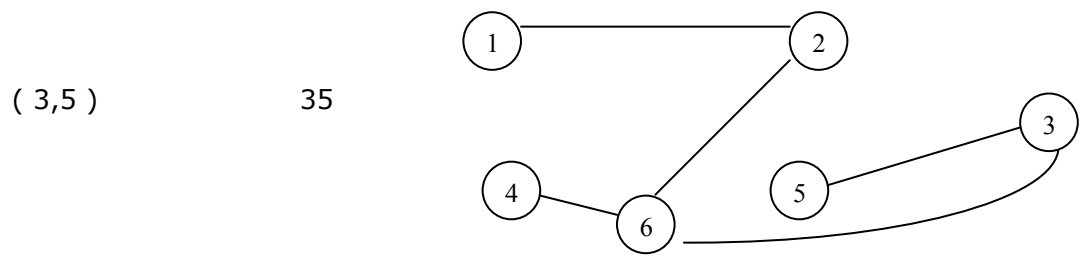
Pandang graph G sebagai berikut :



penyelesaian :

Edge	cost	spanning tree
(1,2)	10	
(3,6)	15	
(4,6)	20	
(2,6)	25	
(1,4)	30	

Tidak diterima (karena tidak membentuk tree)



_____ +

Total Cost : 105

Jadi, minimum spanning tree dari graph G adalah 105.