

# **PERTEMUAN 13**

## **Penyelesaian Dengan Algoritma Pemrograman Greedy**

## 3. Penyelesaian Dengan Algoritma Pemrograman Greedy

Algoritma GREEDY KNAPSACK

PROCEDURE GREEDY\_KNAPSACK (W, x, n)

float W[n], x[n], M, isi;

int i, n;

$x(1 : 1) \leftarrow 0$  ; isi  $\leftarrow M$  ;

FOR i  $\leftarrow 1$  TO n

{

IF  $W(i) > \text{isi}$

EXIT

ENDIF

$x[i] \leftarrow 1$

isi  $\leftarrow \text{isi} - W[i]$

}

IF  $i \leq n$ ;

$x[i] \leftarrow \text{isi} / W[i]$

ENDIF

END\_GREEDY\_KNAPSACK

## ***Algoritma Pemrograman Greedy*** (Lanjutan)

Efektif jk data ( $P_i/W_i$ ) disusun secara tidak naik lebih dahulu.

**Penyelesaiannya :**

*Dengan Algoritma Pemrograman Greedy.*

Diket. bhw kapasitas  **$M = 20\text{kg}$** , degan jumlah brg  **$n=3$**

Berat  $W_i$  masing<sup>2</sup> barang = ( $W_1, W_2, W_3$ ) = (18, 15, 10)

Nilai  $P_i$  masing<sup>2</sup> barang = ( $P_1, P_2, P_3$ ) = (25, 24, 15)

Lakukan pengurutan secara tdk naik terhadap hasil  $P_i/W_i$ , misalnya :

$P_1/W_1 \rightarrow 25/18 = 1,39$  menjadi urutan ke 3

$P_2/W_2 \rightarrow 24/15 = 1,60$  menjadi urutan ke 1

$P_3/W_3 \rightarrow 15/10 = 1.50$  menjadi urutan ke 2

Sehingga m'hasilk' pola urutan data yg baru,yaitu

$W_1, W_2, W_3 \rightarrow 15, 10, 18$  dan

$P_1, P_2, P_3 \rightarrow 24, 15, 25$

## Algoritma Pemrograman Greedy (Lanjutan)

Lalu data<sup>2</sup> tsb diinputk' pd Alg. Greedy, terjadi proses :

$x(1:n) \leftarrow 0$  ; isi  $\leftarrow 20$  ;  $i = 1$

$W(i) > \text{isi} ? \rightarrow 15 > 20 ? \rightarrow$  kondisi SALAH

$x(1) = 1 \rightarrow$  b'arti bhw brg tsb dpt dimuat seluruhnya.

Isi =  $20 - 15 \rightarrow$  kapasitas ransel b'kurang dengan  
sisa 5kg  $i = 2$

$W(2) > \text{isi} ?? \rightarrow 10 > 5 ?? \rightarrow$  kondisi BENAR

$x(2) = 5/10 = 1/2 \rightarrow$  benda 10kg hanya dpt dimuat 1/2 bagian  
yaitu 5 kg.

$i = 3$

Endif  $\rightarrow$  diakhiri krn ransel sdh penuh (max = 20kg)

Profit nilai yang didapat adalah :  $P1 + P2 + P3$  yaitu:

$$24.1 + 15.1/2 + 18.0 = 24 + 7.5 = 31.5$$

# Algoritma Pemrograman Greedy (Lanjutan)

## Penyelesaiannya:

$x(1:n) \leftarrow 0$  ; isi  $\leftarrow 20$  ;  $i = 1$

FOR  $i \leftarrow 1$  TO 3

$W1, W2, W3 \rightarrow 15, 10, 18$   
dan

$P1, P2, P3 \rightarrow 24, 15, 25$

Saat  $i=1$  APK  $W[1] > \text{isi}$ ?  $\rightarrow 15 > 20$ ?

$x[1] \leftarrow 1$  barang dapat dimuat seluruhnya

$\text{isi} = 20 - 15 = 5$  sisa kapasitas 5kg

Saat  $i=2$  APK  $W[2] > \text{isi}$ ?  $\rightarrow 10 > 5$ ?  $\rightarrow \text{exit}$

APK  $i \leq n$ ?  $\rightarrow 2 \leq 3$ ?

$x[2] = \text{isi}/W[2] = 5/10 = 1/2$  benda 10kg  
dimuat  $\frac{1}{2}$  bag = 5

ENDIF diakhiri karena ransel sudah penuh (max =20kg)

**Profit nilai :  $P1 + P2 + P3$  yaitu:**

$$24(1) + 15(1/2) + 18(0) = 24 + 7,5 = 31,5$$

# Kesimpulan Knapsack Problem

- Cara matematika dianggap lebih rumit dan tidak cocok untuk digunakan, karena harus memperhatikan nilai probabilitas setiap item, nilai ini merupakan faktor penentu mengingat nilai probabilitas ( $X_i$ )  $0 \leq X_i \leq 1$ . Kisaran nilai-nilai  $X_i$  di sini sangat luas, bisa 0, 0,1, 0,01, 0,001, ... 1.
- Cara kriteria greedy dianggap lebih mudah dan lebih optimal dibanding cara yang lain meskipun kekurangannya harus mengerjakan beberapa tahapan terlebih dahulu.
- Cara algoritma greedy lebih cepat penyelesaiannya namun harus tahu algoritma dan harus paham cara penterjemahan algoritma tersebut. Selain itu teknik ini akan efektif jika objek disusun secara tidak naik terlebih dahulu berdasarkan nilai  $P_i/W_i$ .

# PROBLEMA DAN MODEL GRAPH DALAM METODE GREEDY

## 1. TRAVELLING SALESMAN

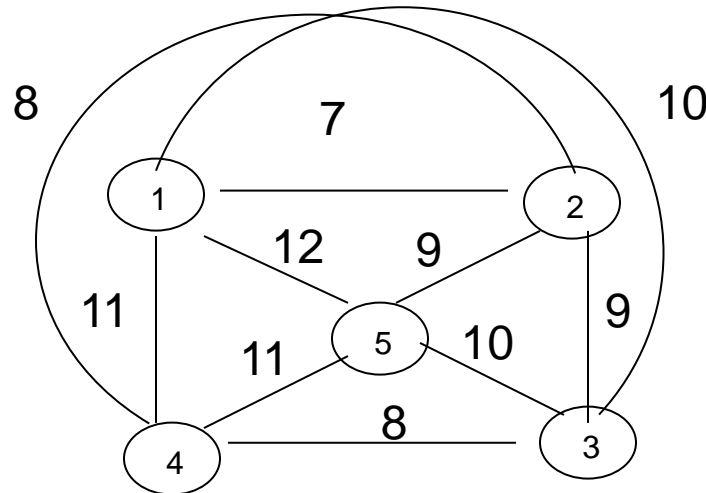
Untuk menentukan waktu perjalanan seorang salesman seminimal mungkin.

### **Permasalahan:**

Setiap minggu sekali, seorang petugas kantor telepon berkeliling untuk mengumpulkan coin-coin pada telepon umum yang dipasang diberbagai tempat. Berangkat dari kantornya, ia mendatangi satu demi satu telepon umum tersebut dan akhirnya kembali ke kantor lagi. Masalahnya ia menginginkan suatu rute perjalanan dengan waktu minimal.

# TRAVELLING SALESMAN (Lanjutan)

## MODEL GRAPH



### Misalnya :

Kantor pusat adalah simpul 1 dan misalnya ada 4 telepon umum, yang kita nyatakan sebagai simpul 2, 3, 4 dan 5 dan bilangan pada tiap-tiap ruas menunjukkan waktu ( dalam menit ) perjalanan antara 2 simpul .



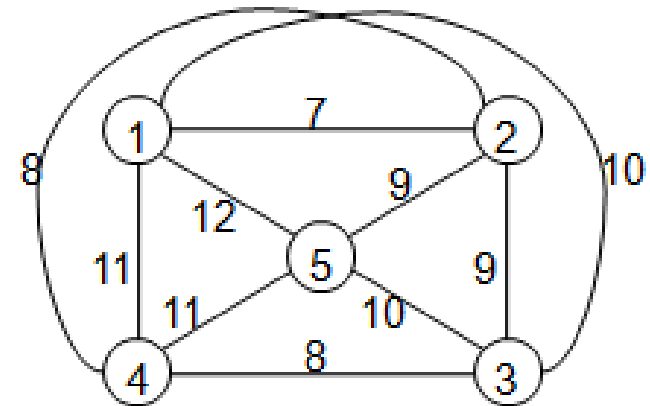
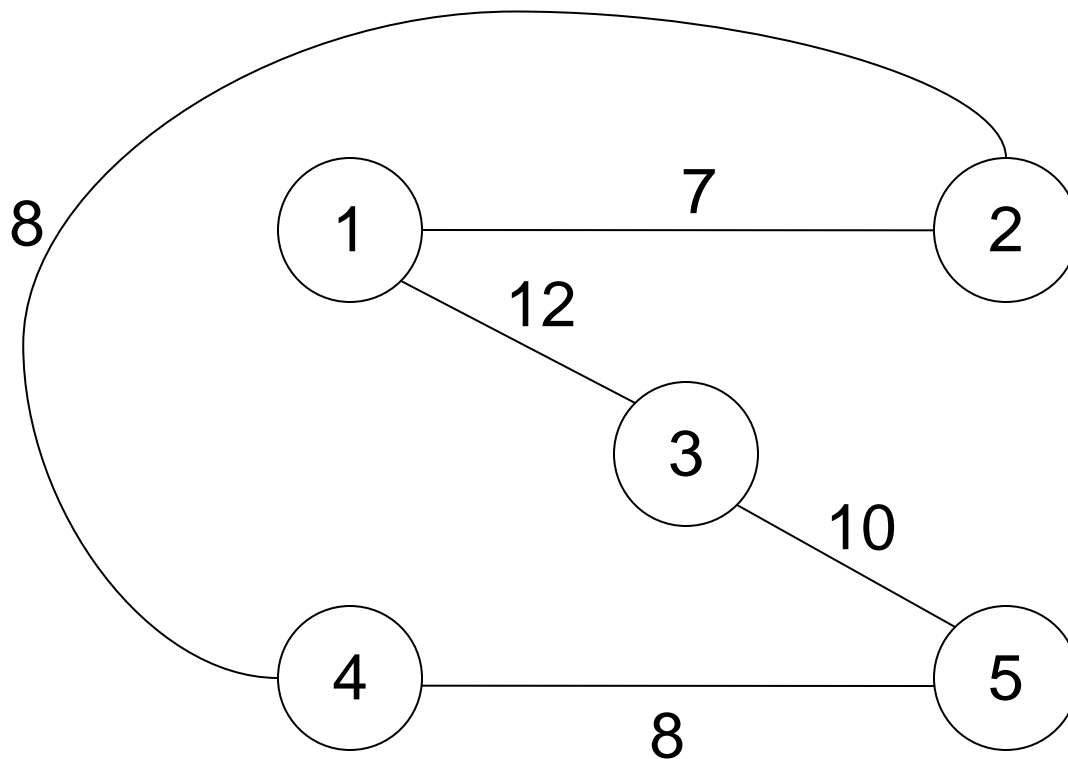
# TRAVELLING SALESMAN (Lanjutan)

## Langkah penyelesaian:

1. Dimulai dari simpul yang diibaratkan sebagai kantor pusat yaitu simpul 1
2. Dari simpul 1 pilih ruas yang memiliki waktu yang minimal.
3. Lakukan terus pada simpul–simpul yang lainnya tepat satu kali yang nantinya Graph akan membentuk Graph tertutup karena perjalanan akan kembali ke kantor pusat.
4. Problema diatas menghasilkan waktu minimalnya adalah 45 menit dan diperoleh perjalanan sbb :

# TRAVELLING SALESMAN (Lanjutan)

Problema diatas menghasilkan waktu minimalnya adalah **45** menit dan diperoleh perjalanan sebagai berikut :



## 2. MINIMUM SPANNING TREE

### Kasus MST Problem

Mencari minimum biaya (*cost*) spanning tree dari setiap ruas (*edge*) graph yang membentuk pohon (*tree*).

### Solusi dari permasalahan ini:

- a. Dengan memilih ruas suatu graph yang memenuhi kriteria dari optimisasi yang menghasilkan biaya minimum.
- b. Penambahan dari setiap ruas pada seluruh ruas yang membentuk graph akan menghasilkan nilai/biaya yang kecil (*minimum cost*).

# MINIMUM SPANNING TREE (Lanjutan)

**Kriteria dari Minimum *Spanning Tree*, yaitu :**

1. Setiap ruas pada graph harus terhubung (*connected*)
2. Setiap ruas pada graph harus mempunyai nilai (*label graph*)
3. Setiap ruas pada graph tidak mempunyai arah (graph tidak berarah)

# MINIMUM SPANNING TREE (Lanjutan)

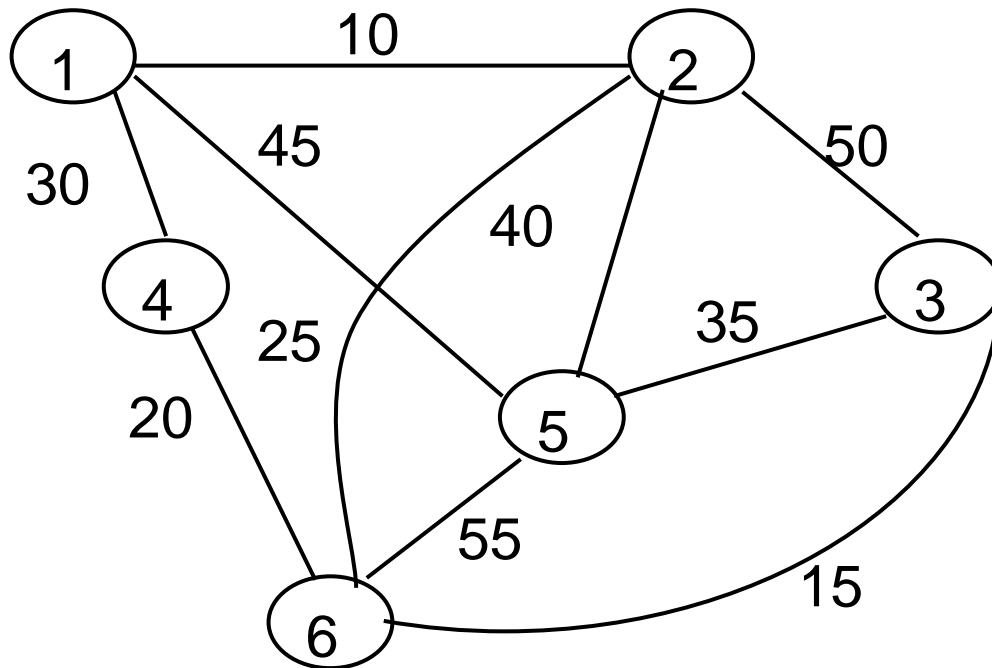
**Proses Total minimum cost terbentuknya graph dengan tahapan sebagai berikut:**

- Dari graph yang terbentuk, apakah memenuhi kriteria MST.
- Lakukan secara urut dari simpul ruas awal s/d ruas akhir
- Pada setiap simpul ruas perhatikan nilai/cost dari tiap-tiap ruas
- Ambil nilai yang paling kecil (jarak terpendek setiap ruas).
- Lanjutkan s/d semua simpul ruas tergambar pada spanning tree
- Jumlahkan nilai/cost yang dipilih tadi.

# MINIMUM SPANNING TREE (Lanjutan)

Kriteria :

- ✓ graph terhubung
- ✓ graph tidak berarah
- ✓ graph mempunyai label



Tentukan nilai MST dari graph di atas serta tentukan ruas (edge) yang membentuk MST

# MINIMUM SPANNING TREE (Lanjutan)

## Penyelesaian MINIMUM SPANNING TREE

**Perhatikan Kriteria dari MST, yaitu:**


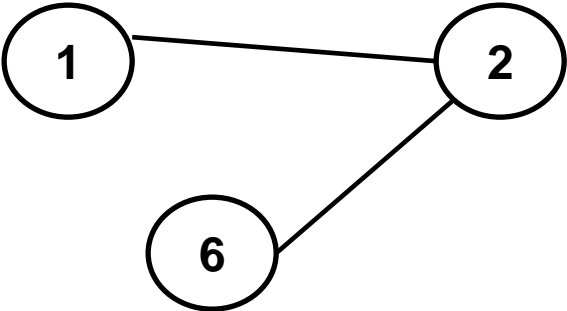
1. Graph sudah merupakan graph terhubung
2. Graph merupakan graph yang tidak berarah
3. Masing-masing ruasnya mempunyai label

**Menghitung MST dari tiap-tiap ruas yang membentuk graph tersebut dengan cara:**

- a. Dilakukan secara urut dari ruas/edge pertama sampai dengan edge terakhir.
- b. Setiap ruas/edge harus digambarkan pada spanning tree yang terbentuk.

# MINIMUM SPANNING TREE (Lanjutan)

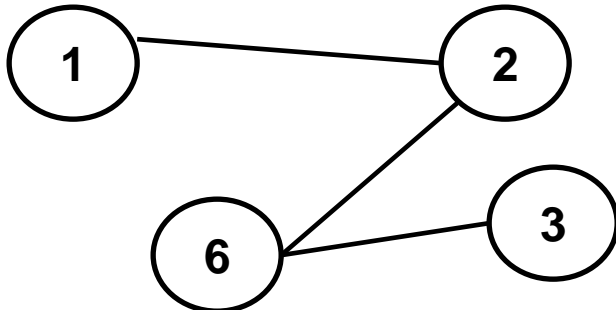
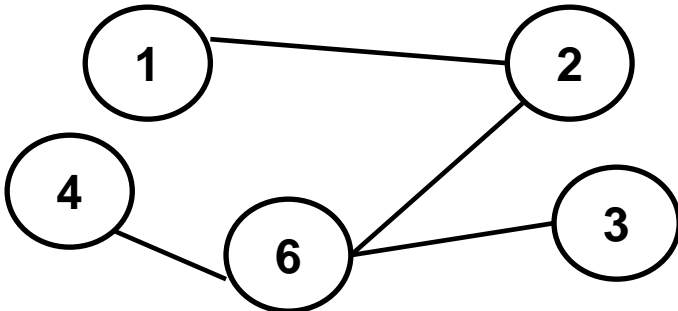
**Tahapan Proses Penyelesaian dari edge (ruas), Cost(biaya) dan spanning tree**

Edge (Ruas)	Cost (Biaya)	Spanning Tree
(1,2)	10	
(2,6)	(25)	



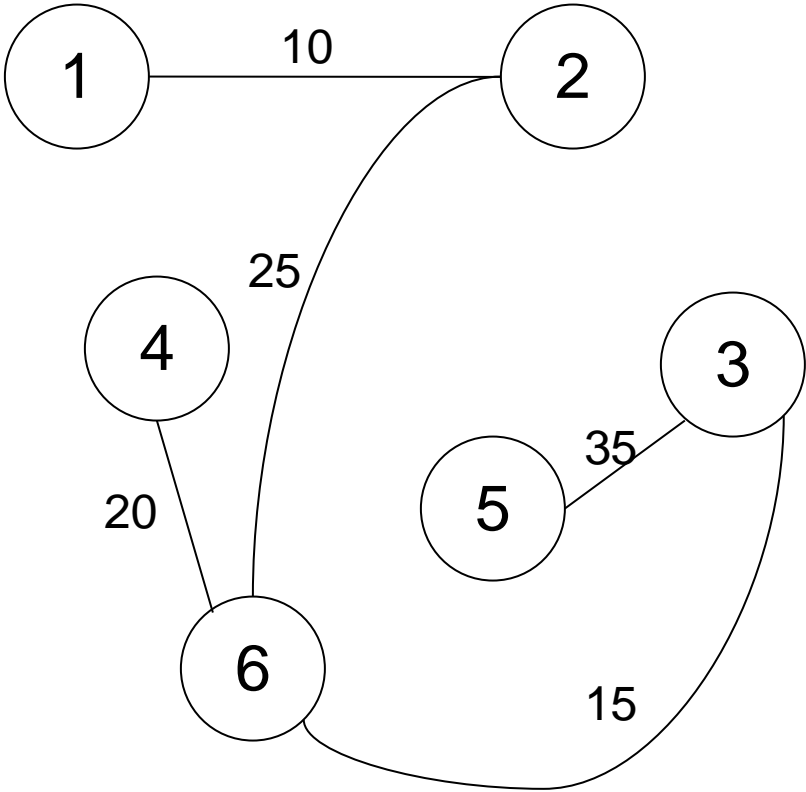
# MINIMUM SPANNING TREE (Lanjutan)

**Tabel Lanjutan Proses Penyelesaian dari edge (ruas), Cost(biaya) dan spanning tree**

Edge (Ruas)	Cost (Biaya)	Spanning Tree
(3,6)	15	
(4,6)	20	

# MINIMUM SPANNING TREE (Lanjutan)

Tabel Lanjutan Proses Penyelesaian dari edge (ruas), Cost(biaya) dan spanning tree

Edge (Ruas)	Cost (Biaya)	Spanning Tree
(3,5)	35	
Total Cost	105	

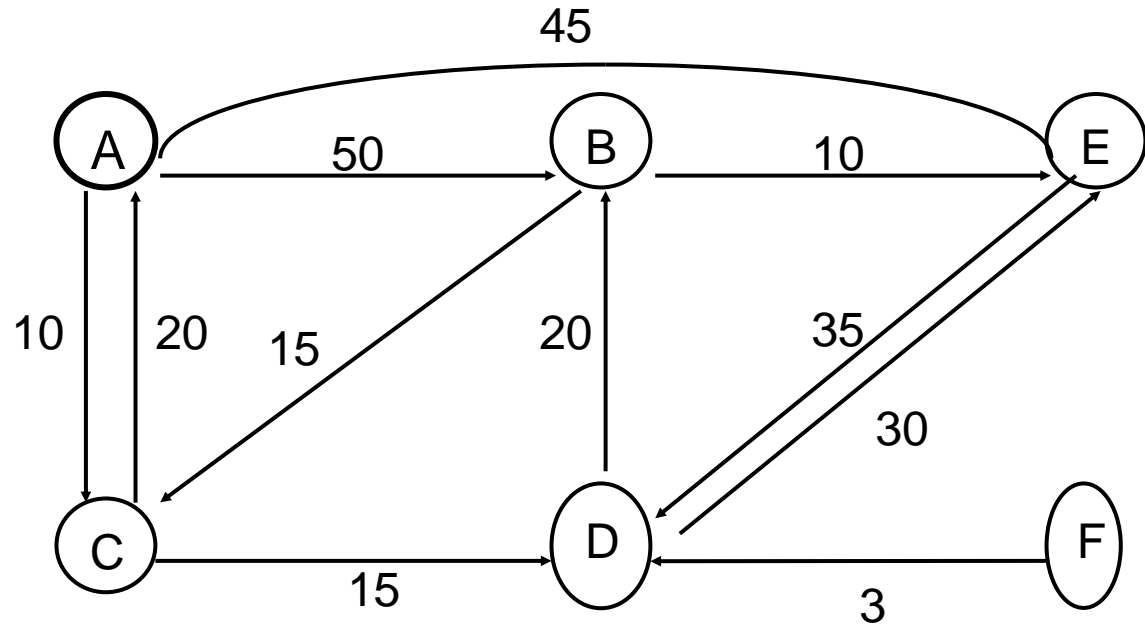
### 3. SHORTEST PATH PROBLEM

**Permasalahan:** Menghitung jalur terpendek dari sebuah graph berarah.

**Kriteria untuk permasalahan Shortest Path problem tersebut :**

1. Setiap ruas pada graph harus mempunyai nilai (*label graph*)
2. Setiap ruas pada graph tidak harus terhubung (*unconnected*)
3. Setiap ruas pada graph tersebut harus mempunyai arah (graph berarah).

# SHORTEST PATH PROBLEM (Lanjutan)



1. Hitung jarak satu per satu sesuai dengan arah yang ditunjukkan oleh tiap-tiap ruas.
2. Perhitungan dilakukan terhadap ruas graph yang memiliki jalur awal dan akhir.

# SHORTEST PATH PROBLEM (Lanjutan)

## Penyelesaian

- Pertama: Melihat proses simpul yang mempunyai awal dan akhir tujuan dari graph, yaitu:  
 $A - B, A - C, A - D, A - E$
- Kedua: Mencari jalur terpendek dari tiap-tiap proses keempat jalur tersebut dengan menghitung panjang tiap-tiap jalur.

# SHORTEST PATH PROBLEM (Lanjutan)

## Langkah 1 Penyelesaian Jalur A - B

- $A - B = 50$
- $A - C - D - B = 10 + 15 + 20 = 45$
- $A - E - D - B = 45 + 30 + 20 = 95$

Jalur terpendek untuk simpul A tujuan B adalah:

$$A - C - D - B = 45$$

# SHORTEST PATH PROBLEM (Lanjutan)

## Langkah 2 Penyelesaian Jalur A - C

- $A - C = 10$
- $A - B - C = 50 + 15 = 65$
- $A - B - E - D - B - C = 50 + 10 + 30 + 20 + 15 = 125$
- $A - E - D - B - C = 45 + 30 + 20 + 15 = 110$

Jalur terpendek untuk simpul A tujuan C adalah:

$$A - C = 10$$

# SHORTEST PATH PROBLEM (Lanjutan)

## Langkah 3 Penyelesaian Jalur A - D

- $A - C - D = 10 + 15 = 25$
- $A - B - E - D = 50 + 10 + 30 = 90$
- $A - B - C - D = 50 + 15 + 15 = 80$
- $A - E - D = 45 + 30 = 75$

Jalur terpendek untuk simpul A tujuan D adalah:

$$A - C - D = 25$$



# SHORTEST PATH PROBLEM (Lanjutan)

## Langkah 4 Penyelesaian Jalur A - E

- $A - E = 45$
- $A - B - E = 50 + 10 = 60$
- $A - C - D - B - E = 10 + 15 + 30 + 10 = 65$
- $A - C - D - E = 10 + 15 + 35 = 60$

Jalur terpendek untuk simpul A tujuan E adalah

$$A - E = 45$$

# SHORTEST PATH PROBLEM (Lanjutan)

## Tabel Jalur SHORTEST PATH PROBLEM

Jalur	Panjang jarak
A – C	10
A – C – D	25
A – C – D – B	45
A – E	45