



# Pertemuan 15

## Graf (Graph)

Tim Ajar Algoritma dan Struktur Data 2021



# Tujuan

- Mahasiswa memahami definisi Graf dan terminologinya
- Mahasiswa mampu memodelkan permasalahan di dunia nyata menggunakan Graf
- Mahasiswa mampu merepresentasikan struktur data Graf

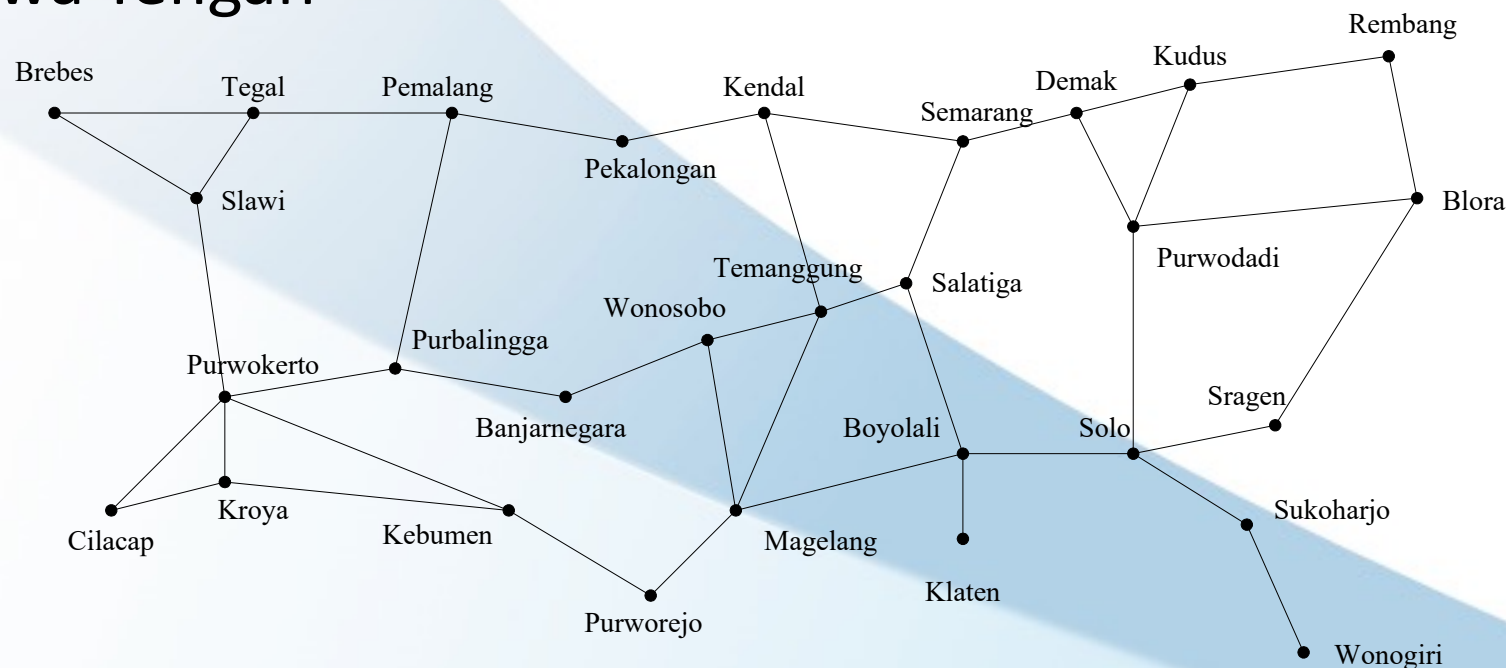


# Outline

- Graf, sejarah Graf, Definisi istilah pada Graf
- Contoh Graf
- Representasi Graf

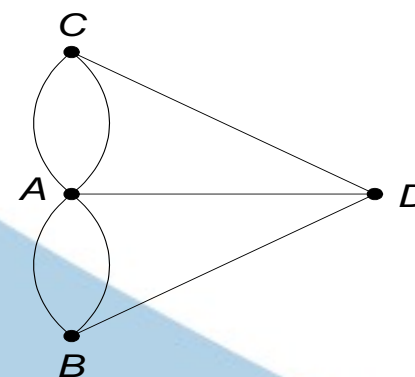
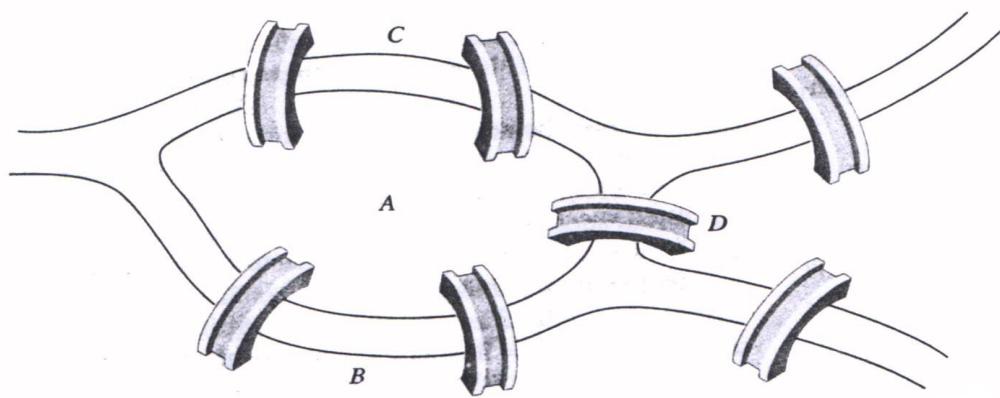
# Graf

- Graph digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut
- Gambar di bawah ini merepresentasikan jalan dan jarak antar kota di provinsi Jawa Tengah



# Sejarah Graf

- Jembatan Königsberg tahun 1973
- Graf yang merepresentasikan jembatan Königsberg:  
*vertex* (titik) → menyatakan daratan  
*edge* (tepi/ garis) → menyatakan jembatan



Bisakah melalui setiap jembatan tepat sekali dan kembali lagi ke titik semula?

# Definisi Graf

Graph  $G = (V, E)$  adalah suatu sistem himpunan berhingga tak kosong  $V(G)$  dan himpunan  $E(G)$  (mungkin kosong) yang elemen-elemennya merupakan himpunan pasangan tak berurut 2 elemen berbeda dari  $V(G)$

Graph  $G = (V, E)$ , yang dalam hal ini:

$V$  = himpunan tidak-kosong dari titik-titik (*vertices*)  
=  $\{a, b, \dots, v_n\}$

$E$  = himpunan garis (*edges*) yang menghubungkan titik-titik  
=  $\{e_1, e_2, \dots, e_n\}$  atau  $\{(a,b) (a,c) (n, n)\}$

# Istilah pada Graf

- ***Vertex* (Titik atau simpul)**

Titik dalam *graph* disebut dengan *vertex*. Biasanya disimbolkan dengan bentuk lingkaran .

- ***Edge* (Garis atau sisi atau tepi)**

Garis-garis penghubung antar titik dalam *graph* disebut dengan garis (*edge*)

- ***Adjacency* (Bertetangga)**

Dua titik (*vertex*) dinamakan bertetangga (*adjacent*) jika saling terhubung melalui satu garis (*edge*).

- ***Path* (Lintasan)**

Path atau lintasan adalah representasi sebuah jalan dari satu titik ke titik lainnya.

# Contoh

## Graf G

$v_1, v_2, v_3, v_4, v_5, v_6$  adalah titik

$e_1, e_2, e_3, e_4, e_5, e_6$  adalah garis

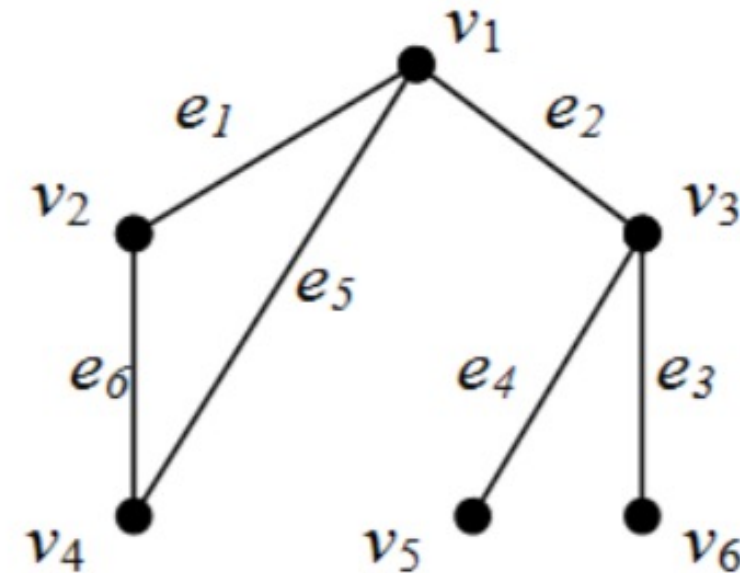
$v_1$  bertetangga dengan  $v_2, v_3$  dan  $v_4$

$v_2$  tidak bertetangga dengan  $v_3, v_5$  dan  $v_6$

Path dari  $v_4$  ke  $v_6$  yaitu  $v_4 \rightarrow v_2 \rightarrow v_1 \rightarrow v_3 \rightarrow v_6$

Path dari  $v_4$  ke  $v_6$  bisa juga  $v_4 \rightarrow v_1 \rightarrow v_3 \rightarrow v_6$

Path terkecil biasa disebut *the shortest path*



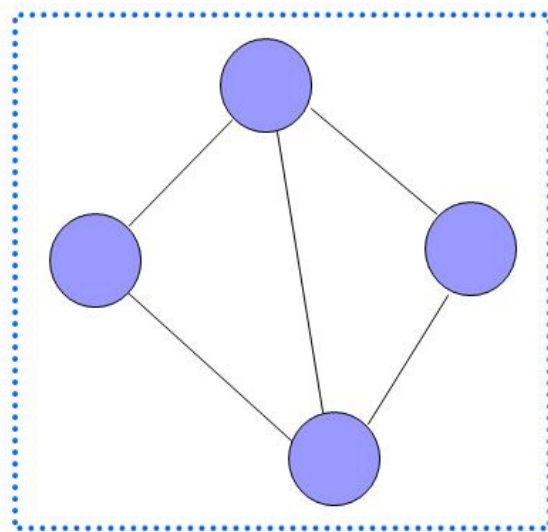
Graf G



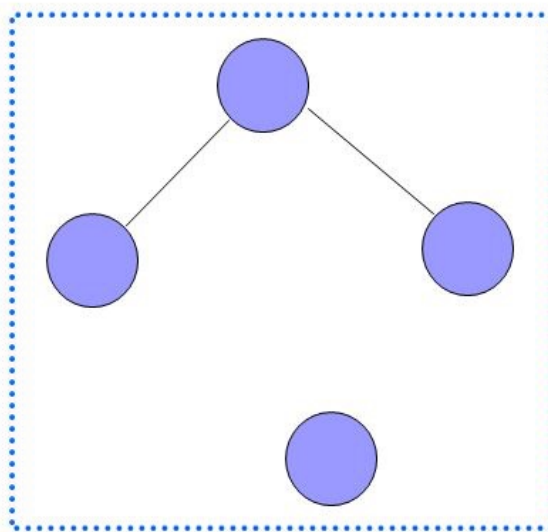
# Istilah pada Graph

- **Terhubung (*Connected*)**

Suatu graph dikatakan *connected* jika ada setidaknya satu garis (*edge*) antara satu titik (*vertex*) ke titik lainnya. Gambar 1 adalah contoh *connected graph*. Sedangkan *graph* tidak terhubung (*unconnected graph*) jika satu atau lebih titik-titiknya tidak terhubung ke titik-titik lainnya.



Connected Graph

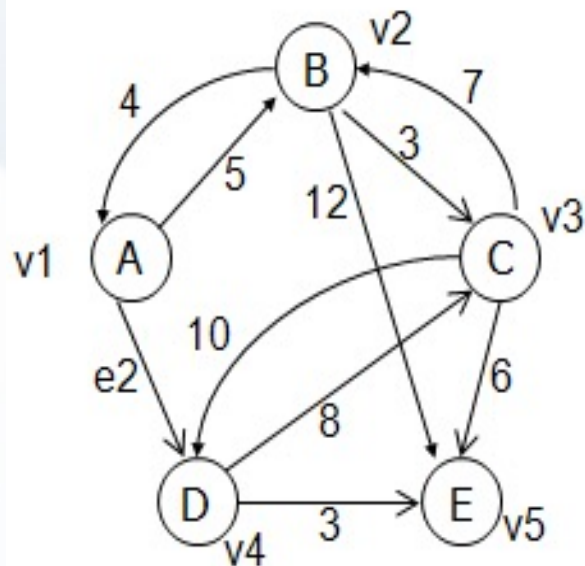


Unconnected Graph

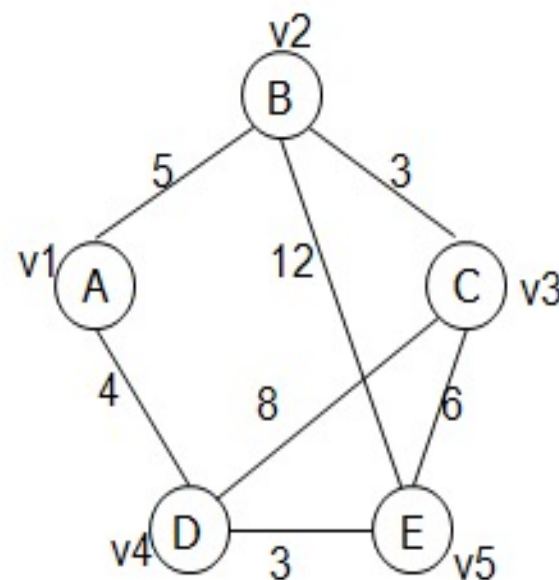
# Istilah pada Graph

- ***Directed Graph*** dan ***Weighted Graph***

*Directed* and *weighted Graph* adalah *graph* dengan garis antar *titik* yang memiliki arah dan mempunyai bobot.



Directed graph



Undirected graph

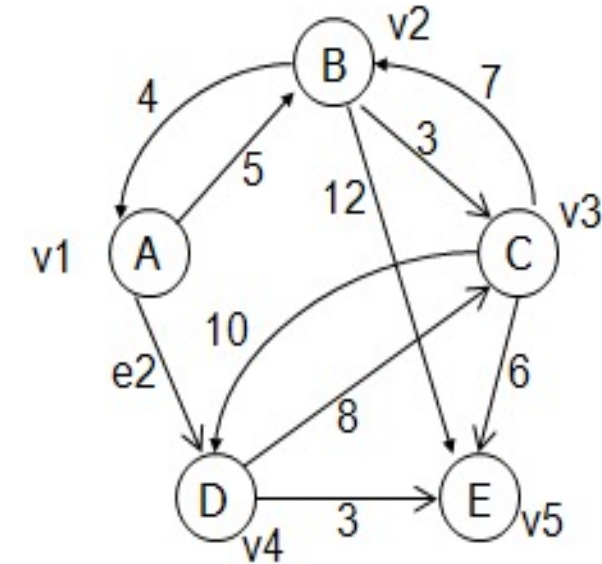
# Istilah pada Graph

- **Degree (derajat), in-degree dan out-degree**

*Degree* sebuah titik adalah jumlah busur yang *incident* (terkait) dengan titik tersebut.

Atau jumlah garis yang terkait dengan titik tersebut.

- **In-degree** sebuah titik pada *graph* berarah adalah jumlah busur yang kepalanya *incident* dengan titik tersebut, atau jumlah busur yang “masuk” atau menuju titik tersebut.
- **Out-degree** sebuah titik pada *graph* berarah adalah jumlah busur yang ekornya *incident* dengan titik tersebut, atau jumlah busur yang “keluar” atau berasal dari titik tersebut.
- Dinotasikan  $d(v)$



Directed graph

$$D_{in}(A) = 1$$

$$D_{out}(A) = 2$$

# Jenis representasi graph

- **Adjacency list**

Adjacency list menggunakan suatu array pada linked list. Array tersebut digunakan untuk menyimpan jumlah *vertex*. Nilai pada linked list dapat digunakan untuk menyimpan bobot graph.

- **Adjacency matrix**

*Adjacency matrix* merupakan array 2D dengan size  $V \times V$  dimana  $V$  adalah jumlah titik pada *graph*. Jika  $adj[i][j] = 1$  dapat diartikan terdapat suatu garis (*edge*) pada titik  $i$  ke titik  $j$ .

# *Adjacency list undirected graph*

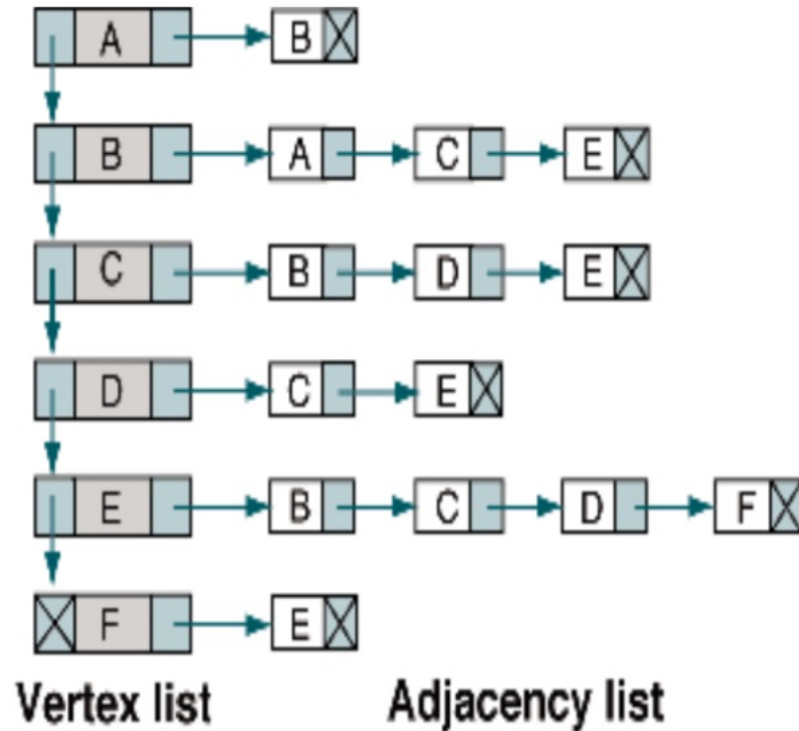
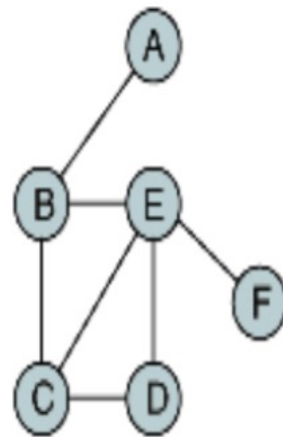
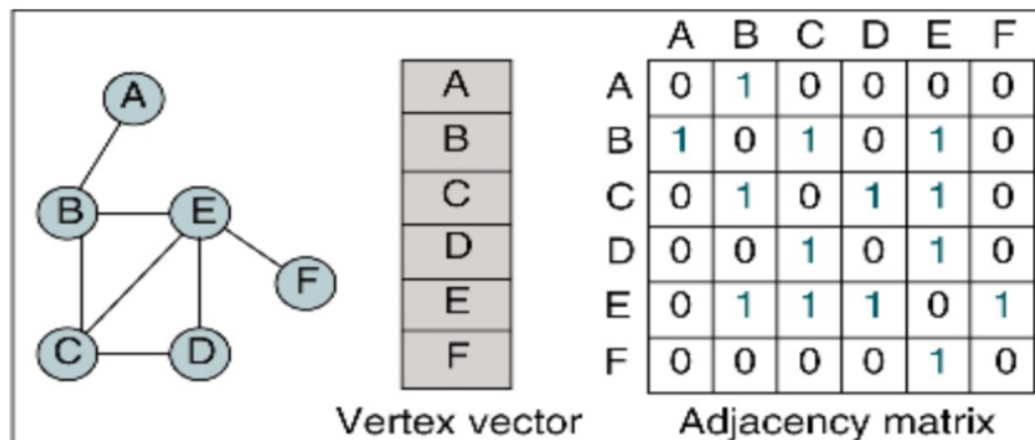
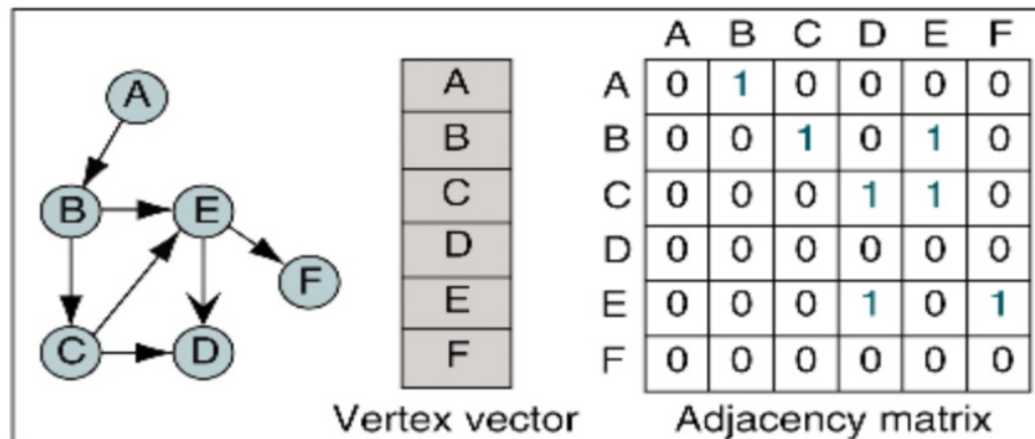


FIGURE 11-14 Adjacency List

# Graph dan matrix adjacency directed graph



**(a) Adjacency matrix for nondirected graph**

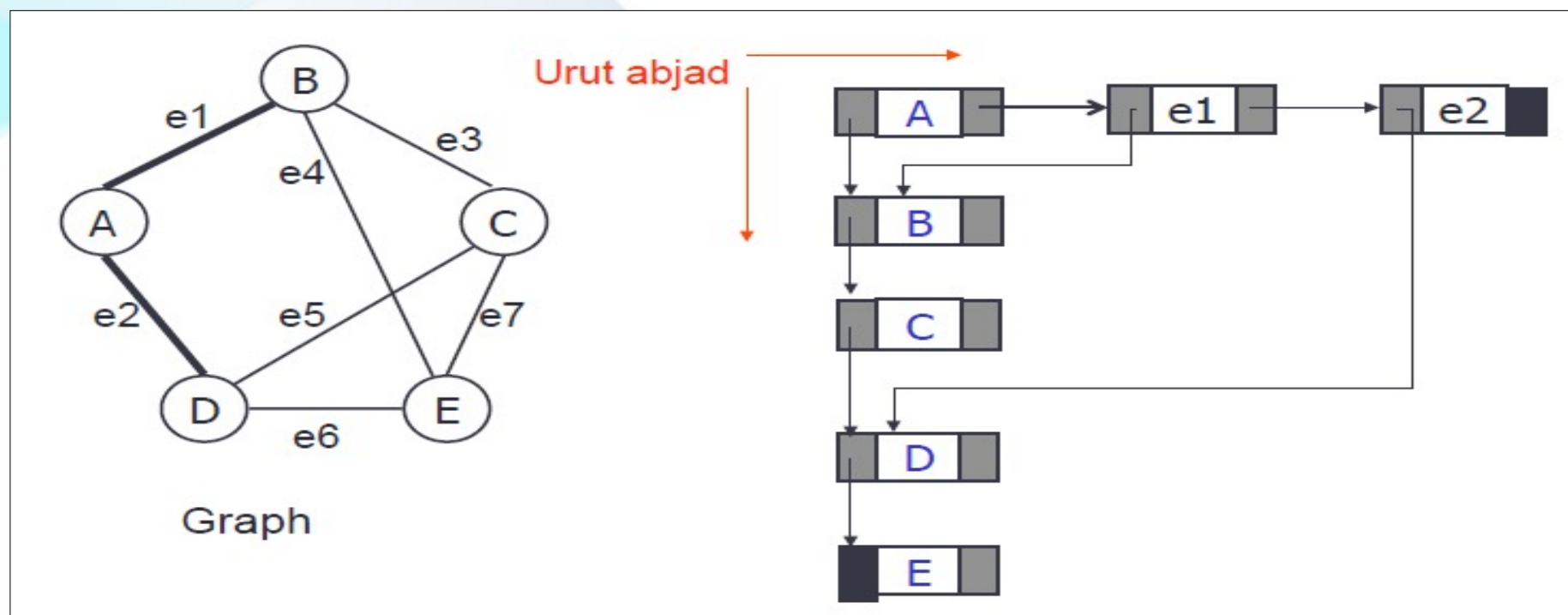


**(b) Adjacency matrix for directed graph**



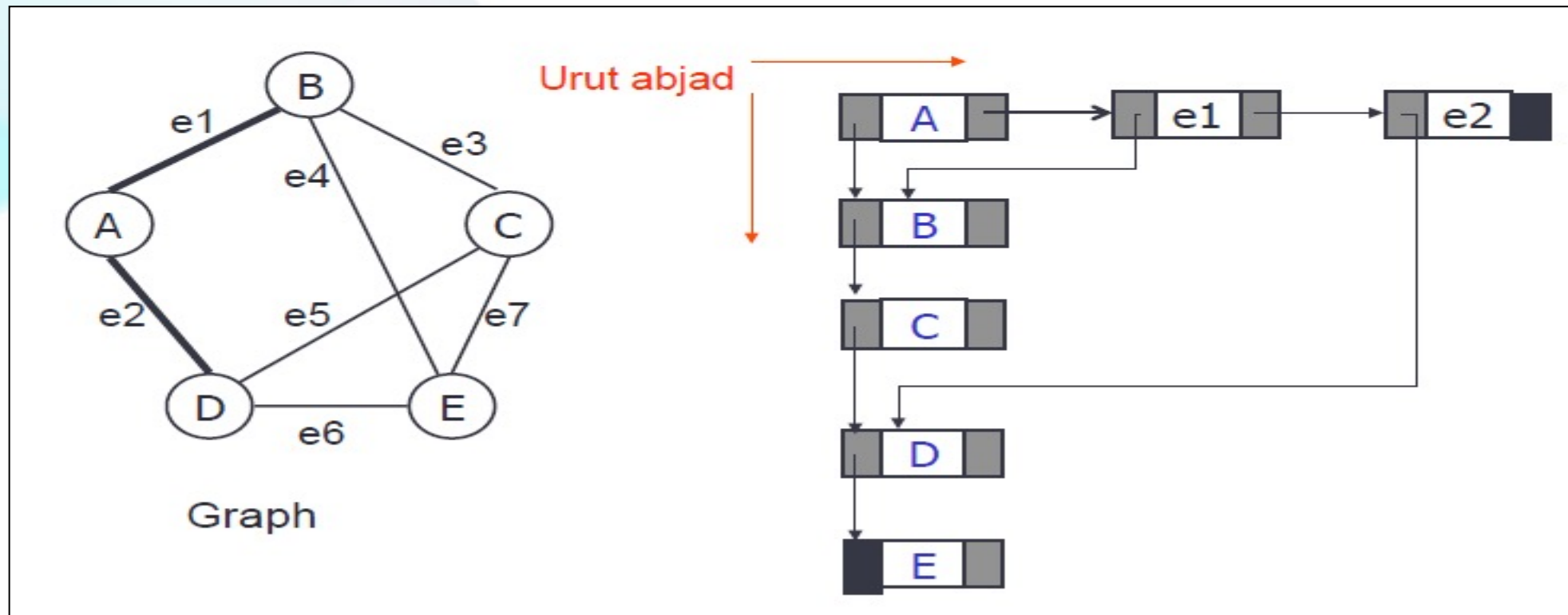
# Representasi Graph dalam bentuk Linked List

- Adjacency List graph tak berarah/berarah
- Digambarkan sebagai sebuah vertex yang memiliki 2 pointer (pointer titik dan pointer garis)



# Contoh(1)-Adjacency Undirected Graph

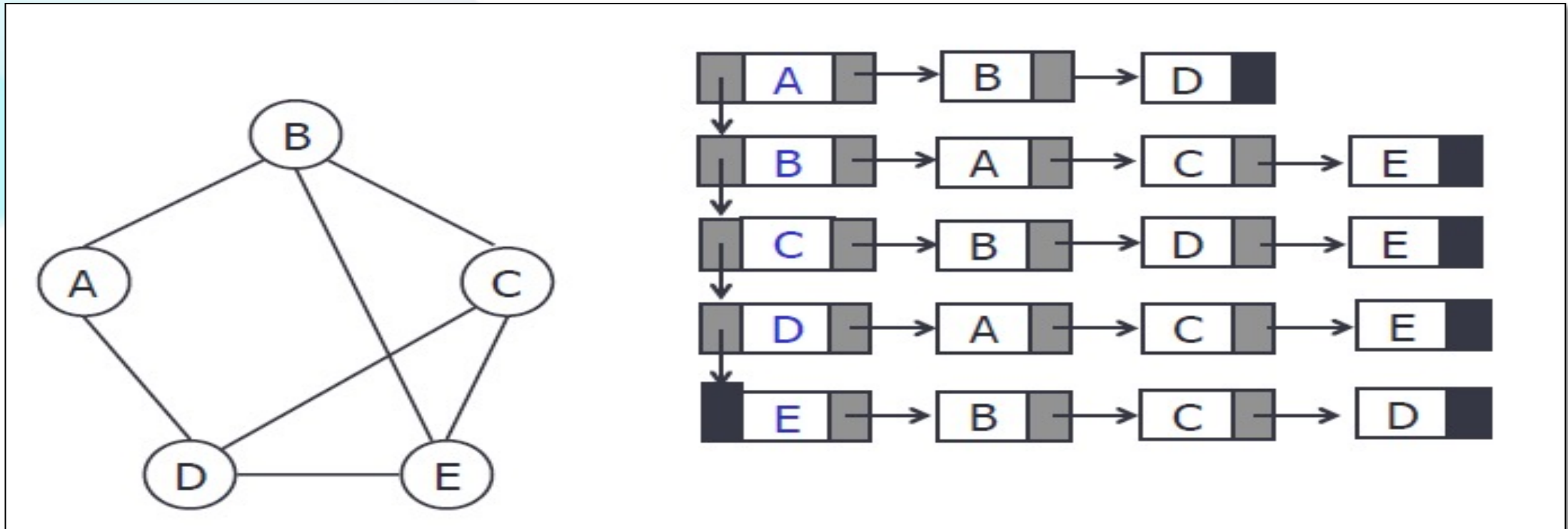
- untuk vertex A, memiliki 2 garis yang terhubung yaitu  $e_1$  dan  $e_2$



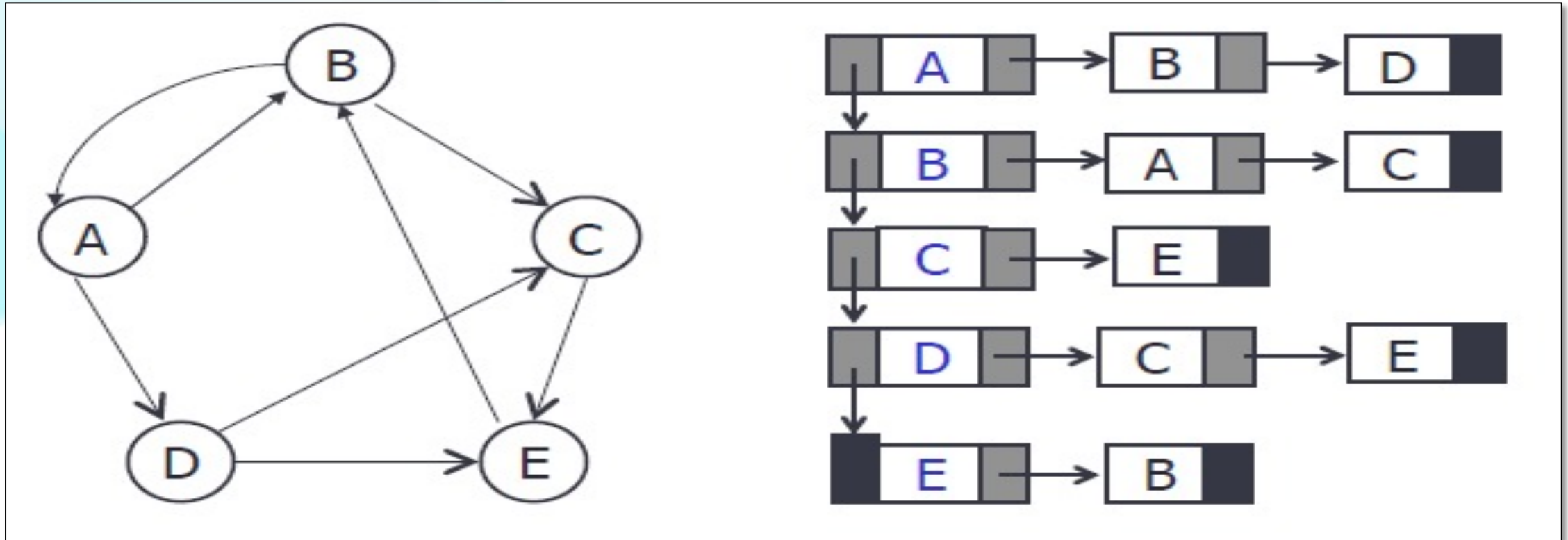


# Contoh(1)

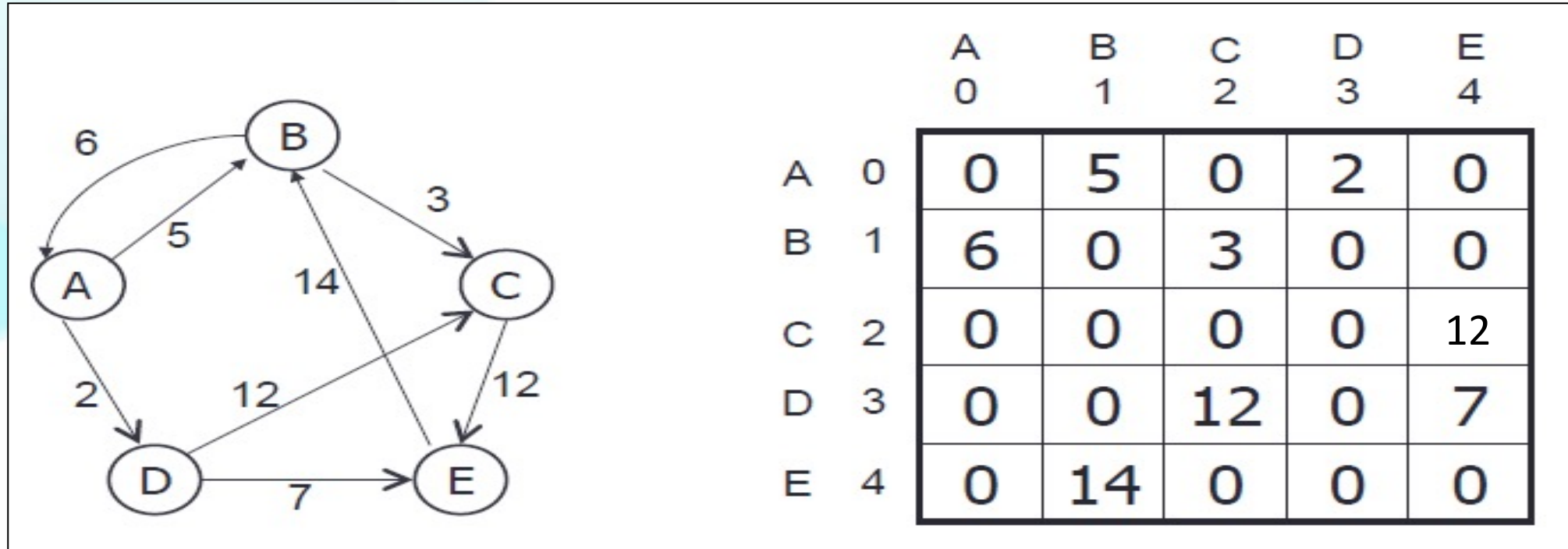
- Bentuk yang lebih sederhana dari contoh 1



## Contoh(2)-Adjacency Directed Graph

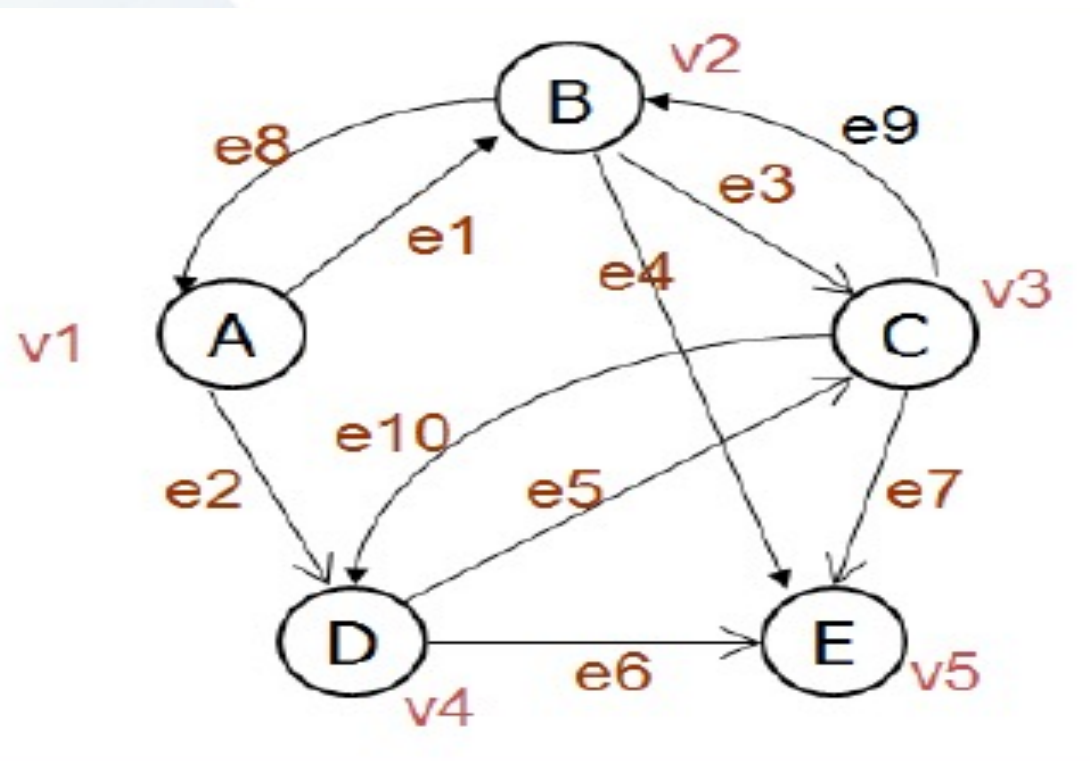


## Contoh(3)-Directed and Weighted Graph



# Latihan 1

- Ubahlah Graf berikut ke dalam bentuk matriks



# Latihan 2

- Ubahlah matriks berikut ke dalam bentuk Graf

	V1	V2	V3	V4	V5	V6
V1	0	1	0	0	0	0
V2	1	1	1	0	0	0
V3	0	1	0	1	1	1
V4	0	0	1	0	0	0
V5	0	0	1	0	0	0
V6	0	0	1	0	0	0

# Latihan 3

- Ubahlah matriks berikut ke dalam bentuk Graf

	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$
V1	1	1	0	1	1	0	0	0
V2	1	0	1	0	0	0	0	0
V3	0	1	1	0	0	1	1	0
V4	0	0	0	1	0	1	0	1
V5	0	0	0	0	0	0	0	1