

VERSI 2.2  
DESEMBER, 2020



# [STRUKTUR DATA]

MODUL 6, GRAPH

TIM PENYUSUN: - DOSEN  
- DICKY PRABOWO OCTIANTO

PRESENTED BY: LAB. TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MALANG

## [STRUKTUR DATA]

**CAPAIAN PEMBELAJARAN MATA KULIAH**

Mahasiswa mampu menguasai & menjelaskan konsep dari graph

**SUB CAPAIAN PEMBELAJARAN MATA KULIAH**

Mahasiswa mampu memahami:

## 1. Graph

**PERSYARATAN PEMAHAMAN**

## 1. Graph

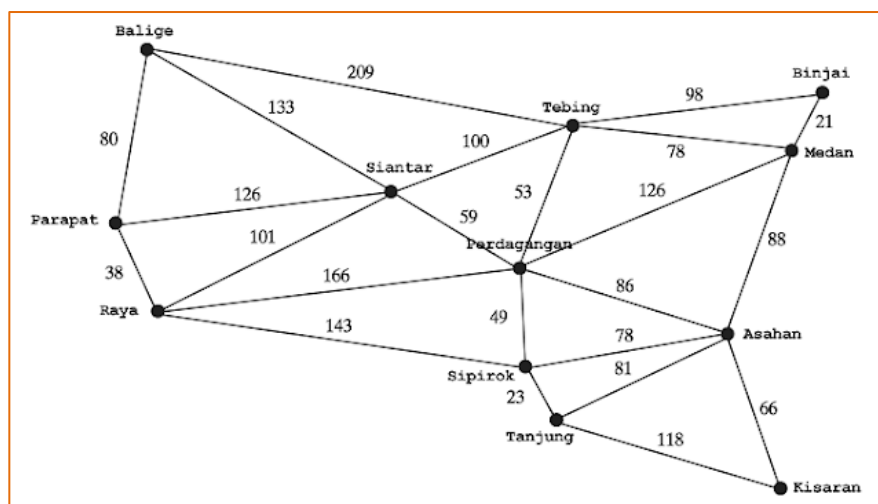
**KEBUTUHAN HARDWARE & SOFTWARE**

- Java Development Kit
- Java Runtime Environment
- IDE (Intellij IDEA, Eclipse, Netbeans, dll.)

**MATERI POKOK**

## 1. Graph

Graph adalah struktur data dengan relasi many to many, yaitu tiap element dapat memiliki 0 atau lebih dari 1 cabang. Graph terdiri dari 2 komponen yaitu Node (digunakan untuk menyimpan data) dan Edge (cabang, untuk menghubungkan node satu dengan node lain). Representasi visual dari graph adalah dengan menyatakan objek sebagai noktah, bulatan atau titik (Vertex), sedangkan hubungan antara objek dinyatakan dengan garis (Edge). Lebih jelasnya dapat dilihat pada gambar dibawah:



1.1 Graph yang menunjukkan jarak antar kota

$$G = \{V, E\}$$

Dimana :

G = Graph

V = Simpul atau Vertex, atau Node, atau Titik

E = Busur atau Edge, atau arc

### 1.2 Representasi Graph

## 2. Jenis-Jenis Graph

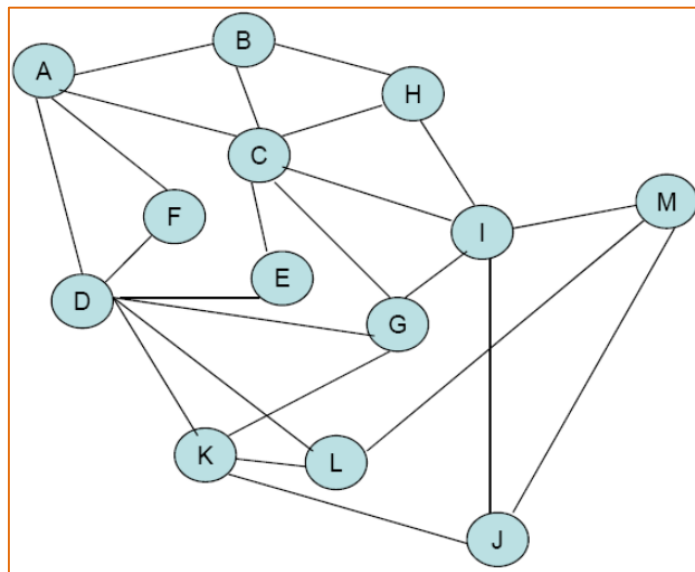
### a. Undirected Graph

Undirected Graph merupakan graph yang tidak memiliki arah, sehingga dalam penulisannya dapat berlaku 2 arah atau bolak balik. Singkatnya perhatikanlah gambar berikut ini untuk dapat lebih memahami :



Dari graph diatas arahnya dapat dituliskan menjadi {A,B} atau {B,A}

Contoh lain adalah sebagai berikut:



Notasi dari gambar diatas:

$G = \{V, E\}$

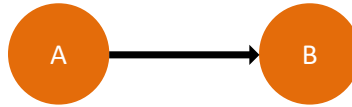
$V = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$

$E = \{ \{A,B\}, \{A,C\}, \{A,D\}, \{A,F\}, \{B,C\}, \{B,H\}, \{C,E\}, \{C,G\}, \{C,H\}, \{C,I\}, \{D,E\}, \{D,F\}, \{D,G\}, \{D,K\}, \{D,L\}, \{E,F\}, \{G,I\}, \{G,K\}, \{H,I\}, \{I,J\}, \{I,M\}, \{J,K\}, \{J,M\}, \{L,K\}, \{L,M\} \}$

## b. Directed Graph

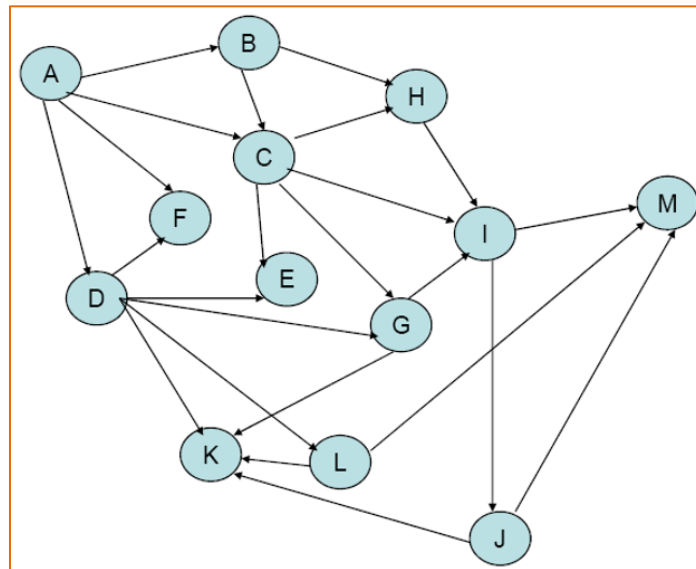
Dari graph diatas arahnya dapat dituliskan menjadi  $\{A,B\}$  dan bukan  $\{B,A\}$

Berbeda dengan Undirected graph yang tidak memiliki arah, direted graph memiliki arah yang tertuju pada node/vertex tertentu yang biasa dinotasikan dengan gambar anak panah. Untuk Lebih Jelasnya dapat memperhatikan gambar berikut:



Dari graph diatas arahnya dapat dituliskan manjadi  $\{A,B\}$  **bukan**  $\{B,A\}$

Contoh lain adalah sebagai berikut:



Notasi dari gambar diatas:

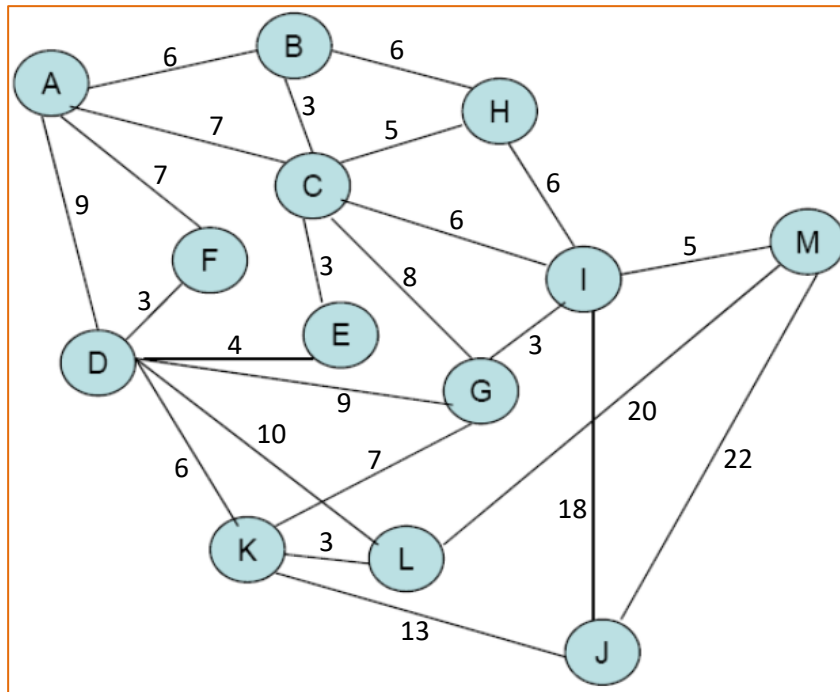
$G = \{V, E\}$

$V = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$

$E = \{(A,B), (A,C), (A,D), (A,F), (B,C), (B,H), (C,E), (C,G), (C,H), (C,I), (D,E), (D,F), (D,G), (D,K), (D,L), (E,F), (G,I), (G,K), (H,I), (I,J), (I,M), (J,K), (J,M), (L,K), (L,M)\}$ .

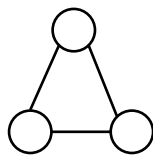
## c. Weight Graph

Sesuai namanya, Weight graph merupakan yang memiliki bobot atau nilai pada tiap – tiap nodenya. Untuk Lebih Jelasnya dapat memperhatikan gambar berikut:



### 3. Istilah – Istilah pada Graph

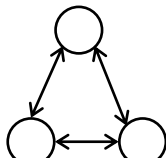
- a. **Jumlah Edge** : Jumlah pasangan edge yang mungkin (banyak maksimal edge) dapat dilihat dari jumlah node ( $n$ ). Dapat dihitung menggunakan formula Undirected Graph :  $n(n-1)/2$  dan untuk Directed Graph :  $n(n-1)$



$n = 3$



Undirected graph  
jumlah maks edge : 3

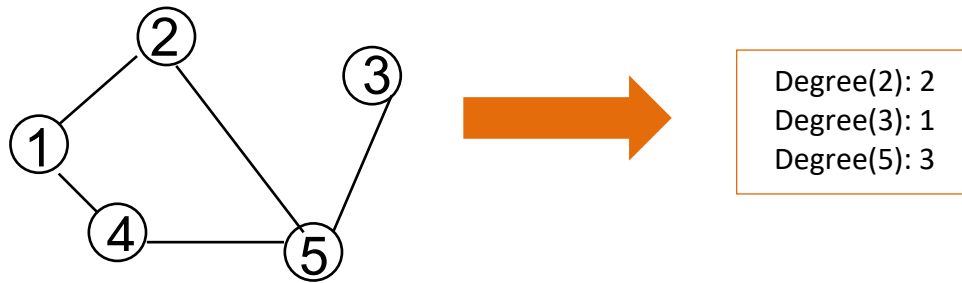


$n = 3$

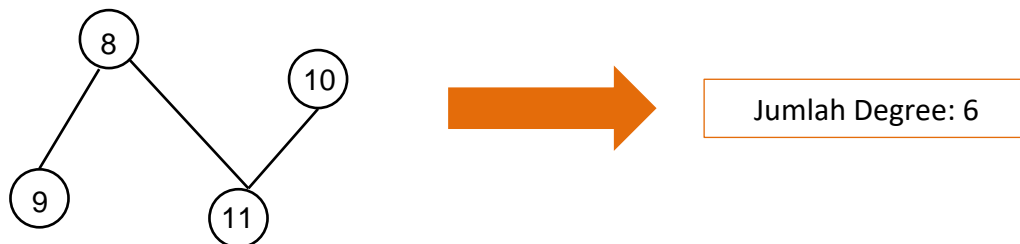


Directed graph  
jumlah maks edge : 6

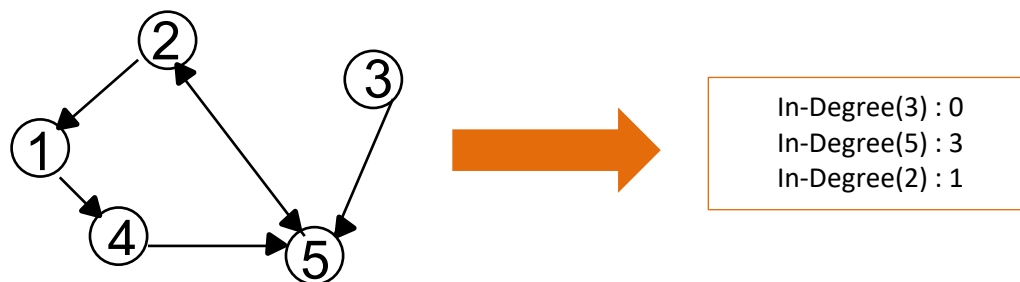
- b. **Degree** : jumlah cabang atau jumlah edge yang dimiliki node



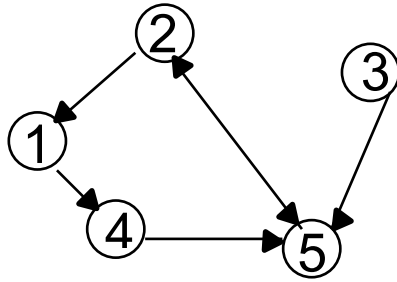
- c. **Jumlah Degree** : Jumlah degree adalah jumlah total cabang/degree yang ada pada graph. Dapat dirumuskan dengan notasi :  $2e$  (e merupakan jumlah edge).



- d. **In-Degree** : Jumlah edge yang masuk atau mengarah ke Node.



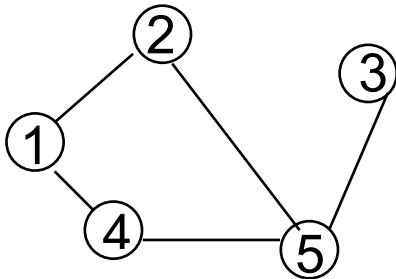
e. **Out-Degree** : Jumlah edge yang keluar dari Node.



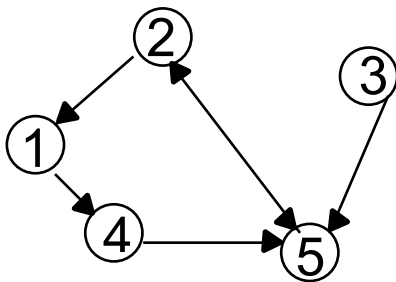
Out-Degree(3) : 1  
 Out-Degree(5) : 0  
 Out-Degree(2) : 1

#### 4. Adjacency Matriks pada graph

Adjacency Matriks pada graph merupakan representasi graph kedalam bentuk matriks dengan ordo  $n \times n$ , dimana  $n$  adalah node. Dalam Adjacency Matriks pada graph ini, baris pada matriks berisi node asal sedangkan kolom pada matriks berisi node tujuan. Jika terdapat edge antar node maka diisi dengan nilai 1 sedangkan jika tidak ada node antar node diisi dengan nilai 0.



	1	2	3	4	5
1	0	1	0	1	0
2	1	0	0	0	1
3	0	0	0	0	1
4	1	0	0	0	1
5	0	1	1	1	0



	1	2	3	4	5
1	0	0	0	1	0
2	1	0	0	0	1
3	0	0	0	0	0
4	0	0	0	0	1
5	0	1	1	0	0

## 5. Graph Traversal

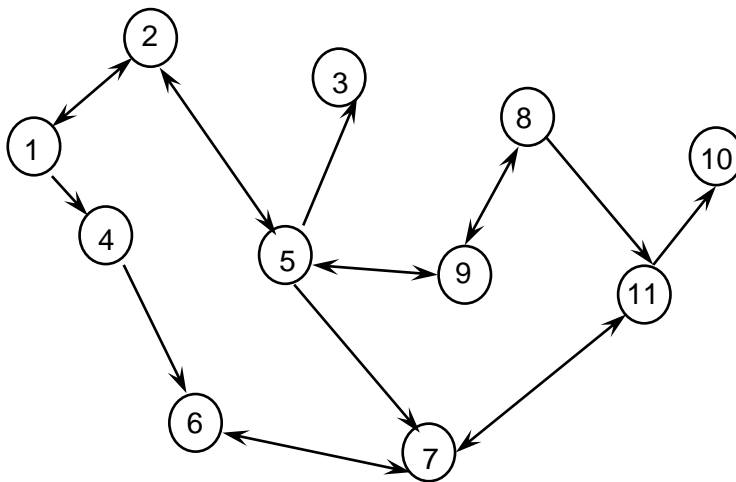
Graph Traversal merupakan metode penelusuran yang digunakan untuk mengunjungi tiap simpul/node secara sistematis. Metode penelusuran pada graph dibagi menjadi 2, yaitu:

- DFS (Depth First Search), melakukan kunjungan ke node-node dengan cara mengunjungi node terdalam/kebawah ,setelah itu mencari tempat yang lainnya , dan sistemnya menggunakan stack.
- BFS (Breadth First Search), melakukan visit ke node- node dengan cara melebar kesamping, dan sistemnya menggunakan queue.

### LEMBAR KERJA

#### KEGIATAN 1

1. Jelaskan kepada asisten tentang apa yang anda ketahui tentang graph dan jelaskan pula perbedaan antara struktur data Tree dan Struktur data Graph.
2. Perhatikan Gambar dibawah



Jelaskan kepada asisten :

- a. Buatlah notasi dari graph diatas (  $G=\{V,E\}$  )
  - b. Tentukan Jumlah in-degree, jumlah out-degree, dan jumlah Edge
  - c. Buatlah Adjacency Matriks dari gambar tersebut
3. Buatlah program yang dapat menentukan DFS (Depth First Search) dan BFS (Breadth First Search) dari gambar pada nomer 2.



---

**CATATAN**

Gunakan library pada Kegiatan 1

**Aturan umum penulisan bahasa JAVA agar mudah di koreksi oleh asisten:**

1. Untuk nama kelas,interface,enum, dan yang lainnya biasakan menggunakan gaya CamelCase (diawali dengan huruf besar pada tiap kata untuk mengganti spasi) seperti: Kursi , JalanRaya, ParkiranGedung, dan lain seterusnya.
2. Untuk penulisan nama method, dan attribute diawali dengan huruf kecil di awal kata dan menggunakan huruf besar untuk kata setelahnya, seperti: getNamaJalan, namaJalan, harga, setNamaJalan, dan lain seterusnya.
3. Jika menggunakan IDE IntelliJ jangan lupa untuk memformat penulisan kode agar terlihat rapi menggunakan menu code -> show reformat file dialog -> centang semua field dan klik ok.

Silahkan dikerjakan tanpa copy - paste

---

**RUBRIK PENILAIAN**

Soal	Nilai
Kegiatan 1:	100%