

# BAB X GRAF

## Pengertian Graf

Graf didefinisikan sebagai pasangan himpunan verteks atau titik ( $V$ ) dan edges atau sisi ( $E$ ). Verteks merupakan himpunan berhingga dan tidak kosong dari simpul-simpul (vertices atau node) dan dapat pula ditulis dengan huruf, bilangan asli atau gabungan keduanya  $\{v_1, v_2, \dots, v_n\}$ . Sedangkan edges merupakan himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul dan dapat ditulis  $\{e_1, e_2, \dots, e_n\}$ .

Notasi graf dapat ditulis  $G = (V, E)$ .

Derajat suatu simpul  $v$  di dalam suatu graf adalah banyaknya simpul yang bertetangga atau terhubung

Notasi :  $V(G)$  = himpunan titik graf  $G$ .       $|V|$  = banyaknya titik pada  $G$

$E(G)$  = himpunan sisi graf  $G$ .       $|E|$  = banyaknya sisi pada  $G$

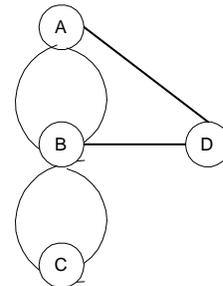
Contoh :  $V(G) = \{A, B, C, D\}$

$E(G) = \{(A,B), (B,A), (A,D), (B,D), (B,C), (C,B)\}$

$|V| = 4$

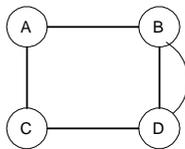
$|E| = 6$

$\text{Deg}(A) = 3$     $\text{Deg}(B) = 5$     $\text{Deg}(C) = 2$     $\text{Deg}(D) = 2$



Multiple edge merupakan beberapa edge bisa mempunyai end point yang sama.

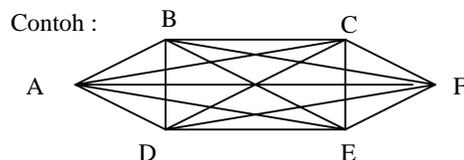
Contoh : Pada gambar graf berikut ini yang merupakan multiple edge adalah (B,D)



Loop merupakan edge yang mempunyai end point pada verteks yang sama. Pada gambar graf dsamping yang merupakan loop adalah verteks C.

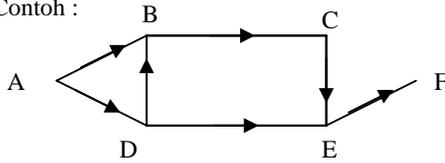
## Jenis-Jenis Graf

1. Complete Graph (Graf Lengkap), suatu graf dikatakan graf lengkap jika semua verteks yang ada dihubungkan ke verteks yang lain.



2. Directed Graph (Graf Berarah), suatu graf dikatakan graf berarah jika garf tersebut mempunyai arah (flow).

Contoh :



Dengan memperhatikan banyaknya jumlah anak panah yang masuk pada satu titik terdapat dua istilah yang lain yaitu :

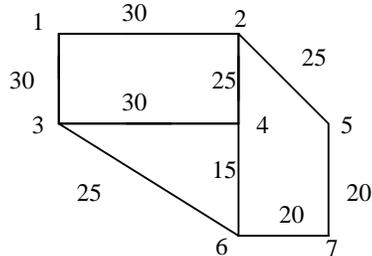
- a. indegree, merupakan banyaknya anak panah yang masuk ke suatu titik.  
 Contoh : indegree dari contoh graf berarah diatas adalah indeg (A) = 0, indeg (B) = 2, indeg (C) = 1, indeg (D) = 1, indeg (E) = 2, indeg (F) = 1.
- b. outdegree, merupakan banyaknya anak panah yang meninggalkan ke suatu titik.  
 Contoh : outdegree dari contoh graf berarah diatas adalah outdeg (A) = 2, outdeg (B) = 1, outdeg (C) = 1, outdeg (D) = 2, outdeg (E) = 1, outdeg (F) = 0.

**Representasi Graf**

Cara merepresentasikan graf antar lain menggunakan :

1 Larik yaitu matrik tetangga (adjency matrix), dimana simpul yang terhubung di dalam matrik pada posisi baris dan kolomnya bernilai 1, jika simpul tidak berhubungna maka di dalam matrik pada posisi baris dan kolomnya bernilai 0.

Contoh 1 :



Pada gambar contoh 1 merupakan graf berbobot dimana setiap edges mempunyai nilai atau bbot. Jika graf tersebut direpresentasikan dalam bentuk matrik, maka yang menjadi baris dan kolomnya adalah jumlah nodenya sehingga menghasilkan matrik n x n. Nilai pada setiap baris dan kolom dari simpul yang berhubungan adalah nilai bobot dari edgesnya. Sedangkan jika simpul tidak berhubungan, maka nilai pada baris dan kolom simpul tersebut di beri nilai 0. Keterhubungan masing-masing simpul dapat bolak balik atau dua arah karena graf tersebut bukan garaf berarah.

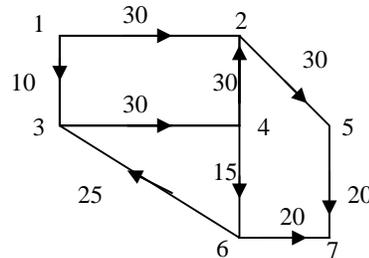
Representasi dalam bentuk matrik

	1	2	3	4	5	6	7
1	0	30	30	0	0	0	0
2	30	0	0	25	25	0	0
3	30	0	0	30	0	25	0
4	0	25	30	0	0	15	0
5	0	25	0	0	0	0	20
6	0	0	25	15	0	0	20
7	0	0	0	0	20	20	0

Representasi dalam bentuk matrik tetangga

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	1	0	0	1	1	0	0
3	1	0	0	1	0	1	0
4	0	1	1	0	0	1	0
5	0	1	0	0	0	0	1
6	0	0	1	1	0	0	1
7	0	0	0	0	1	1	0

Contoh 2 :



Pada gambar contoh 2 merupakan graf berbobot dimana setiap edges mempunyai nilai atau bobot. Jika graf tersebut direpresentasikan dalam bentuk matrik, maka yang menjadi baris dan kolomnya adalah jumlah nodenya sehingga menghasilkan matrik  $n \times n$ . Nilai pada setiap baris dan kolom dari simpul yang berhubungan adalah nilai bobot dari edgesnya. Sedangkan jika simpul tidak berhubungan, maka nilai pada baris dan kolom simpul tersebut di beri nilai 0. Keterhubungan masing-masing simpul tidak dapat bolak balik atau dua arah karena graf tersebut merupakan graf berarah yang tergantung dari arah panah setiap edges .

Representasi dalam bentuk matrik

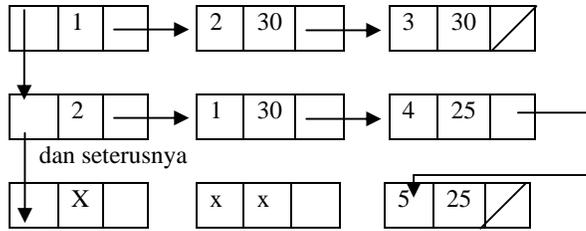
	1	2	3	4	5	6	7
1	0	30	10	0	0	0	0
2	0	0	0	0	30	0	0
3	0	0	0	30	0	0	0
4	0	30	0	0	0	15	0
5	0	0	0	0	0	0	20
6	0	0	0	0	0	0	20
7	0	0	0	0	0	0	0

Representasi dalam bentuk matrik tetangga

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	0	0	0	1	0	0
3	0	0	0	1	0	0	0
4	0	1	0	0	0	1	0
5	0	0	0	0	0	0	1
6	0	0	1	0	0	0	1
7	0	0	0	0	0	0	0

2. Linked list yaitu medan info berisi nilai bobot dari edgesnya, berikut/next menunjuk ke alamat simpul yang terhubung.

Jika graf contoh 1 diatas direpresentasikan menggunakan linked list, maka keadilustrasinya sebagai berikut :



**Aplikasi Graf**

**1. Pencarian Jalur Terpendek (Shortest path)**

Graf digunakan sebagai alat untuk merepresentasikan atau memodelkan suatu persoalan. Berdasarkan graf yang dibentuk barulah persoalan tersebut dapat dianalisis untuk diselesaikan. Salah satu permasalahan yang dapat diselesaikan menggunakan graf adalah mencari lintasan terpendek (*Shortest Path*). Persoalan dalam mencari lintasan terpendek dalam graf merupakan persoalan optimasi klasik. Graf yang digunakan dalam memecahkan permasalahan lintasan terpendek adalah graf berarah dan berbobot. Contoh bagaimana menyelesaikan sebuah kasus dalam suatu jaringan distribusi barang dari distributor ke agen-agennya. Permasalahan yang timbul adalah bagaimana menemukan lintasan terpendek dari simpul asal ke simpul tujuan untuk mendapatkan solusi yang optimal yaitu menghemat waktu komputasi dan biaya.

Salah satu algoritma yang digunakan dalam shortest path ini adalah algoritma Floyd Warshall dimana algoritma tersebut digunakan untuk menemukan lintasan terpendek dan jarak terpendek semua pasangan simpul dalam suatu graf.

**Algoritma Floyd Warshall**

Algoritma ini menyelesaikan pencarian lintasa/rute terpendek ketika fungsi bobotnya tidak dibatasi nilai non negatif. Selain itu juga dapat menyelesaikan masalah lintasan terpendek diantara semua pasangan simpul. Algoritma ini mengijinkan bobot sisi negatif. Langkah-langkah dari algoritma Floyd Warshall adalah sebagai berikut :

Misalkan suatu jaringan dengan  $A = (a_{ij})$  menjadi matriks bobot  $n \times n$  dan  $P = (p_{ij})$  menjadi matriks posisi simpul dengan orde  $n \times n$ , dimana  $n$  adalah banyaknya simpul.

1. Langkah awal

Inisialisasi matriks  $A^{(0)} = A$  dan matriks  $P^{(0)} = P$

$A(i, i) = 0$

$A(i, j) = \sim$ , jika tidak ada sisi dari simpul  $i$  ke simpul  $j$ .

Iterasi tiga diawali dengan dua matriks yaitu  $A^{(c-1)}$  dan  $P^{(c-1)}$  dan iterasi terakhir adalah  $A^{(c)}$  dan  $P^{(c)}$ , dimana  $c = 1, 2, \dots, n$ .

2. Elemen-elemen di dalam matriks dinyatakan sebagai berikut :

a. Jika  $d_{ik}^{(c-1)} \leq d_{ij}^{(c-1)} + d_{jk}^{(c-1)}$  maka, isi matriks  $A^{(c-1)}$  dengan  $(i, k)$  sama dengan isi matriks  $A^{(c)}$  dan isi matriks  $P^{(c-1)}$  dengan  $(i, k)$  sama dengan isi matriks  $P^{(c)}$ . Dimana  $d_{ik}$  merupakan jarak dari simpul  $i$  ke simpul  $k$ ,  $d_{ij}$  merupakan jarak dari simpul  $i$  ke simpul  $j$  dan  $d_{jk}$  merupakan jarak dari simpul  $j$  ke simpul  $k$ .

- b. Jika  $d_{ik}^{(c-1)} \leq d_{ij}^{(c-1)} + d_{jk}^{(c-1)}$ , maka isi (i,k) dalam matriks  $A^{(c)}$  adalah jumlah dari isi (i, j) + isi (j, k) dari matriks  $A^{(c-1)}$  dan isi (i, k) dari matriks  $P^{(c)}$  adalah sama dengan isi (i, j) dalam matriks  $P^{(c-1)}$ .
3. Algoritma ini berhenti jika sudah menemukan simpul tujuan atau  $c = n$  sehingga di dapatkan dua matriks yaitu  $A = A^{(n)}$  dan  $P = P^{(n)}$ .