

Belajaritma

Data Structure

**Sesi VI : Graphs
(PDF Example)**

Pengenalan Graph

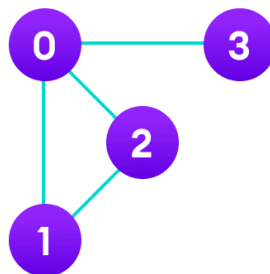
Graphs adalah jenis struktur data yang susunan datanya tidak berdekatan satu sama lain (non-linier). Graph terdiri dari kumpulan simpul berhingga untuk menyimpan data dan antara dua buah simpul terdapat hubungan saling keterkaitan.

Simpul pada graph disebut dengan **verteks (V)**, sedangkan sisi yang menghubungkan antar verteks disebut **edge (E)**. Pasangan (x,y) disebut sebagai edge, yang menyatakan bahwa simpul x terhubung ke simpul y.

Graph memiliki beberapa karakteristik sebagai berikut:

- Jarak maksimum dari sebuah simpul ke semua simpul lainnya dianggap sebagai eksentrisitas dari simpul tersebut.
- Titik yang memiliki eksentrisitas minimum dianggap sebagai titik pusat dari graph.
- Nilai eksentrisitas minimum dari semua simpul dianggap sebagai jari-jari dari graph terhubung.

Sebagai contoh, terdapat graph seperti berikut:



Graph di atas terdiri atas 4 buah verteks dan 4 pasang sisi atau edge. Dengan verteks disimbolkan sebagai V, edge dilambangkan E, dan graph disimbolkan G, ilustrasi di atas dapat ditulis dalam notasi berikut:

$V = \{0, 1, 2, 3\}$

$E = \{(0,1), (0,2), (0,3), (1,2)\}$

$G = \{V, E\}$

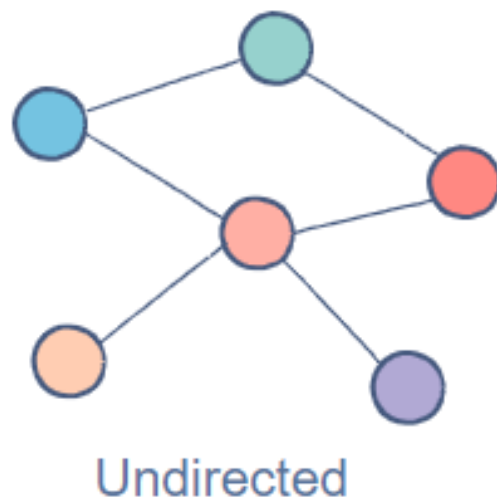
Jenis-jenis Graph

Graph dapat dibedakan berdasarkan arah jelajahnya dan ada tidaknya label bobot pada relasinya.

Berdasarkan arah jelajahnya graph dibagi menjadi **Undirected graph** dan **Directed graph**.

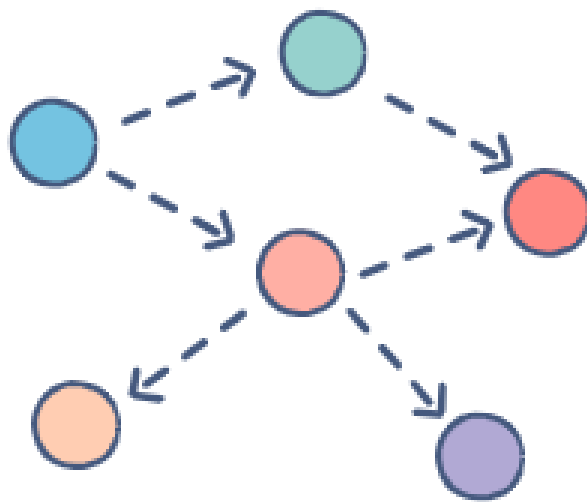
Undirected Graph

Pada undirected graph, simpul-simpulnya terhubung dengan edge yang sifatnya dua arah. Misalnya kita punya simpul 1 dan 2 yang saling terhubung, kita bisa menjelajah dari simpul 1 ke simpul 2, begitu juga sebaliknya.



Directed Graph

Kebalikan dari undirected graph, pada graph jenis ini simpul-simpulnya terhubung oleh edge yang hanya bisa melakukan jelajah satu arah pada simpul yang ditunjuk. Sebagai contoh jika ada simpul A yang terhubung ke simpul B, namun arah panahnya menuju simpul B, maka kita hanya bisa melakukan jelajah (traversing) dari simpul A ke simpul B, dan tidak berlaku sebaliknya.



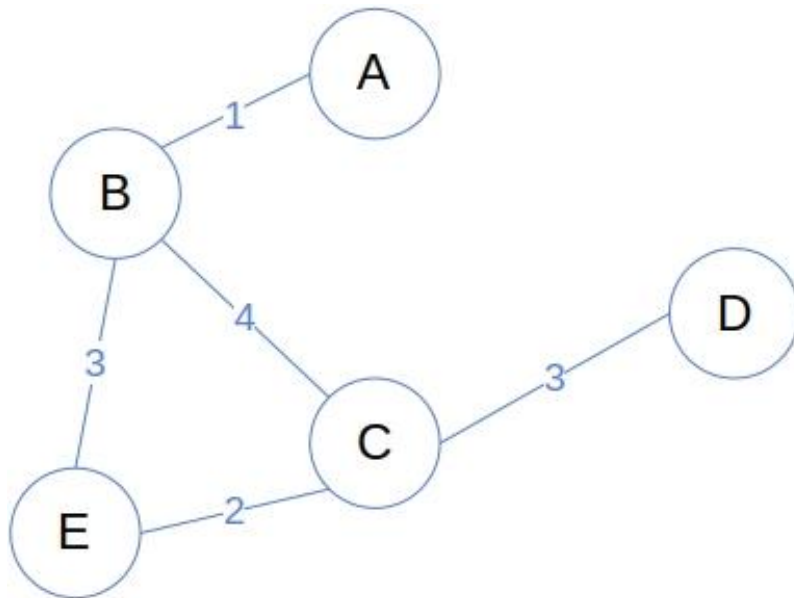
Directed

Selain arah jelajahnya, graph dapat dibagi menjadi 2 berdasarkan ada tidaknya label bobot pada koneksinya, yaitu **weighted graph** dan **unweighted graph**.

Weighted Graph

Weighted graph adalah jenis graph yang cabangnya diberi label bobot berupa bilangan numerik. Pemberian label bobot pada edge biasanya digunakan untuk memudahkan algoritma dalam menyelesaikan masalah.

Contoh implementasinya misalkan kita ingin menyelesaikan masalah dalam mencari rute terpendek dari lokasi A ke lokasi D, namun kita juga dituntut untuk mempertimbangkan kepadatan lalu lintas, panjang jalan dll. Untuk masalah seperti ini, kita bisa mengasosiasikan sebuah edge e dengan bobot $w(e)$ berupa bilangan ril.



Nilai bobot ini bisa apa saja yang relevan untuk masalah yang dihadapi: misalnya jarak, kepadatan, durasi, biaya, probabilitas, dan sebagainya.

Unweighted Graph

Berbeda dengan jenis sebelumnya, unweighted graph tidak memiliki properti bobot pada koneksinya. Graph ini hanya mempertimbangkan apakah dua node saling terhubung atau tidak.

Fungsi dan Kegunaan Graph

Fungsi dan kegunaan graph di antaranya:

- Graph digunakan untuk merepresentasikan aliran komputasi.
- Digunakan dalam pemodelan grafik.
- Graph dipakai pada sistem operasi untuk alokasi sumber daya.
- Google maps menggunakan graph untuk menemukan rute terpendek.
- Graph digunakan dalam sistem penerbangan untuk optimasi rute yang efektif.
- Pada state-transition diagram, graph digunakan untuk mewakili state dan transisinya.
- Di sirkuit, graph dapat digunakan untuk mewakili titik sirkuit sebagai node dan kabel sebagai edge.
- Graph digunakan dalam memecahkan teka-teki dengan hanya satu solusi, seperti labirin.

- Graph digunakan dalam jaringan komputer untuk aplikasi Peer to peer (P2P).
- Umumnya graph dalam bentuk DAG (Directed acyclic graph) digunakan sebagai alternatif blockchain untuk cryptocurrency. Misalnya crypto seperti IOTA