

Nama : Restu Wibisono

NPM : 2340506061

## GRAPH

Graph adalah struktur data yang terdiri dari serangkaian simpul yang terhubung oleh sisi atau edge. Graph digunakan untuk merepresentasikan berbagai jenis relasi antara entitas, seperti jaringan sosial, jaringan jalan, atau hubungan antar objek dalam sistem. Dalam bahasa pemrograman Python, ada beberapa cara untuk merepresentasikan dan memanipulasi graf, di antaranya:

1. Adjacency Matrix: Representasi graf menggunakan matriks yang menunjukkan keterhubungan antar simpul. Setiap entri dalam matriks menunjukkan apakah ada sisi yang menghubungkan dua simpul.
2. Adjacency List: Representasi graf menggunakan daftar yang menyimpan daftar tetangga-tetangga dari setiap simpul. Setiap simpul memiliki daftar yang berisi simpul-simpul yang terhubung langsung dengannya.
3. NetworkX Library: Python memiliki pustaka khusus yang disebut NetworkX yang menyediakan kelas dan fungsi untuk bekerja dengan graf. NetworkX menyediakan berbagai metode untuk membangun, menganalisis, dan memvisualisasikan graf dengan mudah.

Operasi umum yang dapat dilakukan pada graf termasuk penambahan atau penghapusan simpul dan sisi, pencarian jalur atau lintasan, pengujian keterhubungan, analisis jarak, dan sebagainya. Dalam pemrograman Python, graf sering digunakan dalam berbagai aplikasi seperti pemodelan jaringan, analisis jaringan, algoritma pencarian jalur terpendek, algoritma clustering, dan banyak lagi.

Pemahaman tentang representasi graf dan algoritma yang digunakan untuk memanipulasinya penting dalam pengembangan aplikasi yang melibatkan struktur data kompleks ini. Dengan berbagai pilihan representasi dan dukungan pustaka seperti NetworkX, pengembang Python memiliki fleksibilitas dan kemampuan untuk menangani graf dalam berbagai skenario aplikasi.

Graph dalam konteks struktur data memiliki beberapa istilah yang penting untuk dipahami:

1. Node atau Vertex: Merepresentasikan entitas dalam graf. Dalam beberapa kasus, node juga dapat memiliki atribut atau properti tertentu yang menjelaskan karakteristik dari entitas tersebut.
2. Edge atau Sisi: Merepresentasikan hubungan atau koneksi antara dua node dalam graf. Edge biasanya memiliki arah (directed) atau tidak memiliki arah (undirected).
3. Directed Graph: Graf di mana setiap sisi memiliki arah, artinya sisi memiliki titik awal dan akhir. Ini menggambarkan hubungan satu arah antara node-node.
4. Undirected Graph: Graf di mana setiap sisi tidak memiliki arah atau memiliki arah yang tidak jelas. Sisi ini menggambarkan hubungan dua arah antara node-node.
5. Adjacency: Hubungan antara dua node yang terhubung oleh sisi. Dua node yang berdekatan satu sama lain dianggap berdekatan.
6. Adjacency Matrix: Representasi graf menggunakan matriks dua dimensi di mana baris dan kolom mewakili node-node, dan entri matriks menunjukkan apakah ada sisi yang menghubungkan dua node.
7. Adjacency List: Representasi graf menggunakan daftar yang menyimpan daftar tetangga-tetangga dari setiap node. Setiap node memiliki daftar yang berisi node-node yang terhubung langsung dengannya.
8. Path atau Lintasan: Serangkaian node yang terhubung oleh sisi. Path dapat memiliki panjang yang berbeda-beda dan dapat dilalui dari satu node ke node lainnya.
9. Cycle: Path tertutup di mana node awal dan akhir sama. Cycle dapat ditemukan dalam graf yang mengizinkan edge berulang atau dalam graf yang tidak mengizinkan edge berulang.
10. Degree: Jumlah sisi yang terhubung dengan sebuah node. Degree dari sebuah node dalam graf terarah berbeda dengan degree dari sebuah node dalam graf tak terarah.
11. Connected Graph: Graf di mana setiap pasang node terhubung oleh sebuah path. Dalam graf terarah, ada path dari setiap node ke setiap node lainnya.
12. Disconnected Graph: Graf di mana ada setidaknya dua node yang tidak terhubung satu sama lain.

Pemahaman tentang istilah-istilah ini penting dalam memahami konsep dan operasi pada graf dalam pemrograman Python. Istilah-istilah tersebut memberikan dasar untuk memodelkan, menganalisis, dan memanipulasi graf dengan efektif.

Berikut erikut adalah beberapa jenis graf yang umum digunakan dalam struktur data:

1. **Directed Graph (Digraph):** Graf di mana setiap sisi memiliki arah, artinya sisi memiliki titik awal dan akhir. Ini menggambarkan hubungan satu arah antara node-node. Contoh aplikasi dari directed graph adalah jaringan sosial di mana hubungan "mengikuti" adalah unidirectional.
2. **Undirected Graph:** Graf di mana setiap sisi tidak memiliki arah atau memiliki arah yang tidak jelas. Sisi ini menggambarkan hubungan dua arah antara node-node. Misalnya, dalam graf jalan kota, jalan dua arah antara dua persimpangan dapat direpresentasikan sebagai sisi undirected.
3. **Weighted Graph:** Graf di mana setiap sisi memiliki bobot atau nilai terkait dengannya. Bobot ini dapat merepresentasikan jarak, biaya, atau nilai lain yang relevan tergantung pada konteks aplikasi. Graf berbobot sering digunakan dalam algoritma pencarian jalur terpendek atau algoritma optimasi.
4. **Unweighted Graph:** Graf di mana setiap sisi tidak memiliki bobot atau nilai terkait dengannya. Ini dapat dianggap sebagai kasus khusus dari weighted graph di mana setiap sisi memiliki bobot yang sama atau diabaikan sama sekali.
5. **Cyclic Graph:** Graf di mana terdapat setidaknya satu siklus atau lintasan tertutup. Sebuah siklus adalah path tertutup di mana node awal dan akhir sama. Graf ini dapat berupa directed atau undirected.
6. **Acyclic Graph:** Graf di mana tidak terdapat siklus. Ini berarti tidak ada path yang membentuk loop atau lintasan tertutup. Directed acyclic graph (DAG) sangat umum digunakan dalam berbagai aplikasi, seperti pemrograman dinamis dan representasi struktur data hierarkis.
7. **Connected Graph:** Graf di mana setiap pasang node terhubung oleh sebuah path. Dalam graf terarah, ada path dari setiap node ke setiap node lainnya. Jika graf tidak terarah, disebut sebagai strongly connected jika ada path dari setiap node ke setiap node lainnya.
8. **Disconnected Graph:** Graf di mana ada setidaknya dua node yang tidak terhubung satu sama lain. Ini berarti ada setidaknya dua komponen terhubung secara terpisah dalam graf.