

A Study of Graph Theory: Perbedaan antara *Three* dan *Graph*

Alasan Mempelajari *Graph Theory*

Saya merupakan seorang *software engineer* tanpa *CS (Computer Science) background*, yang lahir dari adanya *coding bootcamp*. Kalau boleh bercerita sedikit, latar belakang kuliah saya dari teknik perminyakan. Begini kurang lebih perjalanan saya dari awal hingga akhirnya bisa menjadi *software engineer*:

- SMP dan SMA sangat suka matematika dan fisika, sangat suka soal logika matematika karena *challenging* dan tidak perlu menghafal.
- Sempat 1 semester ambil kuliah teknik informatika, belajar pascal dan C++
- Lanjut kuliah teknik perminyakan di kampus yang berbeda (mulai dari semester 1), Alhamdulillah dapat nilai cukup bagus di matkul *Structured Programming and Database System*
- Kuliah teknik perminyakan lagi untuk S2
- *Coding bootcamp* 3 bulan, baru setelah nya kerja sebagai *full-stack* dan lebih suka dalam hal pengembangan *backend*

Karena saya tidak punya basic CS yang mumpuni, saya merasa bahwa saya bisa develop program tetapi tidak terlalu paham untuk hal-hal yang menyangkut ilmu-ilmu di CS, padahal *algorithm* dan *data structure* sangat penting dalam hal *software development*. Selama pengalaman apply kerja di beberapa perusahaan, baik dalam maupun luar negeri, saya lebih suka soal-soal *coding test* yang lebih fokus ke arah pembuatan program (membuat *CRUD* dengan *framework* tertentu misalnya) ketimbang harus menyelesaikan *coding challenge* (seperti *leetcode* atau *neetcode*). Sayangnya, masih banyak perusahaan mengharuskan *coding test* dan/atau *coding interview* untuk menyelesaikan soal-soal *algorithm and data structure*. Maka dari itu, saya ingin sedikit-sedikit bisa belajar dan memahami konsep-konsep yang lebih *advance* tentang *algorithm and data structure*.

So, *let's learn about graph theory!*

Pendahuluan

Graph Theory merupakan cabang ilmu matematika yang mempelajari struktur dan hubungan antara objek yang disebut "simpul" (*node or vertice*) dan "sisi" (*link or edge*). Dalam pembelajaran ini, saya tertarik untuk memahami konsep dasar teori graf, termasuk perbedaan antara dua struktur yang penting, yaitu *three* dan *graph*.

Definisi *Three*

Sebuah *three* adalah jenis *graph* khusus yang terdiri dari simpul-simpul yang terhubung oleh sisi-sisi. Dalam *three*, setiap simpul memiliki satu simpul pusat yang disebut akar (*root*), dan setiap simpul (kecuali akar) memiliki tepat satu simpul tetangga yang berada di atasnya. Pohon tidak memiliki sirkuit tertutup (siklus), yang berarti tidak ada jalur yang membentuk lingkaran tertutup. Pohon juga merupakan struktur hirarkis yang digunakan untuk merepresentasikan relasi antara objek dalam bentuk cabang dan subcabang.

Definisi *Graph*

Graph, di sisi lain, adalah struktur yang lebih umum daripada *three*. Graf terdiri dari simpul-simpul yang terhubung oleh sisi-sisi, dan simpul-simpul tersebut tidak memiliki batasan khusus dalam hal jumlah tetangga. *Graph* dapat memiliki sisi yang saling terhubung dalam berbagai cara, termasuk hubungan satu-satu, hubungan satu-ke-banyak, atau bahkan hubungan banyak-ke-banyak antara simpul-simpulnya. *Graph* juga dapat memiliki siklus, yang merupakan jalur tertutup yang membentuk lingkaran dalam strukturnya.

Perbedaan Utama

Perbedaan utama antara *Tree* dan *Graph* adalah:

Sirkuit: *Three* tidak memiliki siklus tertutup (*acyclic*), sedangkan graf dapat memiliki siklus (*cyclic*).

Struktur Hirarkis: *Three* memiliki struktur hirarkis dengan akar dan cabang, sementara graph tidak memiliki struktur hirarkis yang terdefinisi secara khusus.

Batasan Tetangga: Setiap simpul dalam *three* memiliki tepat satu simpul tetangga di atasnya, sedangkan simpul-simpul dalam graph tidak memiliki batasan khusus dalam hal jumlah tetangga.

Contoh Penggunaan

Three sering digunakan untuk representasi struktur seperti struktur data *binary tree*, hierarki organisasi, atau pengelompokan klasifikasi. Graph, di sisi lain, digunakan untuk memodelkan hubungan kompleks antara objek, seperti jaringan sosial, jaringan transportasi, atau representasi matematis dari masalah optimasi.

Manfaat Teori Graf:

Saya tertarik mempelajari teori graf karena memiliki banyak aplikasi yang luas dalam berbagai bidang, termasuk ilmu komputer, jaringan, optimasi, bioinformatika, dan banyak lagi. Pemahaman yang kuat tentang teori graf akan memberi saya landasan yang kokoh untuk menghadapi tantangan yang berkaitan dengan analisis, desain, dan pemodelan sistem kompleks.

Struktur Data:

Graph theory memberikan kerangka kerja yang kuat untuk memahami dan menerapkan struktur data. Dengan memahami sifat dan karakteristik *tree* dan *graph*, saya akan dapat memilih dan mengimplementasikan struktur data yang efisien untuk memecahkan masalah yang melibatkan keterkaitan antara objek.

Graph Algorithm:

Teori graf juga melibatkan pengembangan algoritma yang efisien untuk mencari jalur terpendek (*shortest path*), penelusuran graf (*graph traversal*), pemetaan warna (*color mapping*), analisis jaringan, dan banyak lagi. Dengan mempelajari *graph algorithm*, saya akan dapat mengembangkan keterampilan pemecahan masalah dan analisis yang mendalam, yang sangat penting dalam bidang ilmu komputer dan pengembangan perangkat lunak.

Rencana Pembelajaran:

Saya akan mengembangkan rencana pembelajaran yang komprehensif, termasuk membaca artikel dan menonton video penjelasan tentang *graph theory*. Saya juga akan mempraktikkan pemecahan masalah dan menerapkan konsep-konsep teori graf dalam soal-soal coding yang sering ditanyakan dan berhubungan dengan graph theory, contoh:

- Finding the shortest path cell
- Count how many islands
- Rotten tomatoes
- etc

Manfaat di Masa Depan:

Dengan pemahaman yang kuat tentang *graph theory*, saya berharap dapat mengerjakan soal-soal *coding challenge* dengan lebih baik terutama soal-soal yang menerapkan pola dan ilmu tentang *graph theory*. Saya juga menemukan salah satu platform untuk berlatih *coding interview* yaitu Pramp (Practice Makes Perfect): <https://www.pramp.com/>

Kesimpulan

Dalam studi *graph theory*, pemahaman tentang perbedaan antara *three* dan *graph* adalah krusial. *Three* adalah kasus khusus graf yang memiliki struktur hirarkis dan tidak memiliki siklus. Sementara itu, *graph* lebih umum dan dapat memiliki berbagai jenis hubungan dan siklus. Dengan memahami perbedaan ini, saya akan dapat menerapkan konsep-konsep *graph theory* dengan baik dalam pengembangan dan analisis sistem yang melibatkan struktur data grafikal.