

# Pengenalan Struktur Data & Algoritma

Overview & Pentingnya

Fadhel & Iqdam

# Peta Pembelajaran Semester Ini

Berikut adalah topik utama yang akan kita kuasai bersama untuk membangun fondasi yang kuat dalam ilmu komputer:

01

## W1: Pengenalan Python

Memulai perjalanan kita dengan dasar-dasar pemrograman yang esensial.

02

## W2: Object-Oriented Programming (OOP)

Mempelajari paradigma pemrograman yang efisien dan terstruktur.

03

## W3: Big O Notation

Memahami efisiensi algoritma untuk mengoptimalkan kinerja program.

04

## W9: Sorting and Searching

Menguasai berbagai metode pengurutan dan pencarian data.

05

## W10: Hashing

Mempelajari teknik penyimpanan dan pengambilan data cepat.

06

## W11-12: Recursion

Memahami fungsi yang memanggil dirinya sendiri untuk menyelesaikan masalah kompleks.

07

## W13-14: Tree

Mengenal struktur data hierarkis untuk representasi data yang terorganisir.

# Struktur Data & Algoritma: Dasar-dasarnya

Untuk memahami pentingnya, mari kita definisikan kedua konsep kunci ini:

## Struktur Data

Cara mengatur dan menyimpan data secara efisien di dalam komputer. Ini tentang bagaimana kita menata informasi sehingga mudah diakses dan dimanipulasi.

### Analogi: Rak Buku Terorganisir

Bayangkan sebuah perpustakaan. Mencari buku di rak yang rapi dan berlabel jauh lebih cepat daripada mencari di tumpukan buku acak, bukan?

## Algoritma

Serangkaian langkah-langkah logis yang jelas dan terdefinisi untuk menyelesaikan sebuah masalah atau melakukan suatu tugas.

### Analogi: Resep Masakan

Sebuah resep yang jelas memberitahu Anda langkah demi langkah untuk membuat kue yang sempurna. Tanpa resep yang benar, hasilnya mungkin tidak sesuai harapan.

Intinya: Struktur data adalah **wadahnya**, algoritma adalah **cara mengolah isinya**.

# Mengapa Ini Penting? #1: Efisiensi Program

Pilihan struktur data dan algoritma yang tepat dapat membuat perbedaan besar pada kinerja aplikasi Anda, terutama saat menangani volume data besar.

## Contoh Kasus: Aplikasi Sosial Media dengan 1 Juta Pengguna

1

### Algoritma Buruk

**Pencarian Linear:** Untuk mencari 1 pengguna, sistem harus memeriksa satu per satu dari awal. Di kasus terburuk, ini butuh **1 juta operasi**.

2

### Algoritma Baik

**Pencarian Biner** (pada data terurut): Sistem hanya butuh sekitar **20 operasi** untuk menemukan pengguna yang sama.

# Mengapa Ini Penting? #2: Kemampuan Problem-Solving

Mempelajari Struktur Data & Algoritma melatih Anda untuk memodelkan masalah dunia nyata ke dalam solusi komputasi yang efektif.

## Contoh Kasus: Fitur "Rute Terpendek" di Aplikasi Peta (mis. Google Maps)

### Struktur Data: Representasi Peta

- Peta direpresentasikan sebagai **Graph**.
- Kota adalah **titik (node)**.
- Jalan adalah **garis penghubung (edge)** antar kota.

### Algoritma: Pencarian Rute

- Menggunakan **Algoritma Dijkstra** untuk menghitung jalur terpendek.
- Menemukan rute optimal dari Titik A ke Titik B.

---

Ada Pertanyaan?