
LATIHAN 1: IMPLEMENTASI DAN ANALISIS ALGORITMA SORTING

waktu: pertemuan 2, cek batas waktu di e-learning

Tujuan

Tujuan pemberian tugas adalah agar mahasiswa dapat memahami implementasi algoritma sorting, mampu menganalisis kompleksitas waktu, dan memahami efisiensi algoritma berdasarkan situasi tertentu.

Anda diharapkan dapat:

1. Implementasi algoritma sorting.
2. Menganalisis “kualitas” algoritma secara eksperimental.

—————
*Dengan ini, Anda menyatakan bahwa Anda siap menerima segala konsekuensi
jika nantinya ditemukan adanya kecurangan dalam pengerjaan tugas ini.*
—————

Aturan pengerjaan tugas:

1. Latihan dikerjakan secara berkelompok, dengan anggota 3 orang.
2. Latihan boleh diketik/ditulis tangan (pastikan bisa dibaca), boleh menggunakan Bahasa Indonesia/Inggris. Hindari menggunakan tinta merah.
3. Latihan dikumpulkan dalam format pdf. Jika menggunakan tulis tangan, harap discan (tidak difoto), kemudian dikompresi untuk memperkecil ukuran file.
4. Format penamaan tugas: **L1_Kelas_NomorKelompok_NimAnda**.
Contoh: **L1_4A_03_2315101001**. Pastikan penulisan sesuai dengan format.
5. Pengumpulan tugas melalui e-learning Undiksha (lihat format penamaan di e-learning).
6. Pahami dengan baik tujuan tugas ini. Anda harus mengerti dengan baik jawaban yang Anda kumpulkan nanti.
7. Latihan dinilai berdasarkan kejelasan serta kesesuaian jawaban/penjelasan dengan pertanyaan yang diajukan. Keterlambatan dalam pengumpulan tugas mengurangi poin penilaian.

Rincian tugas

1. Pilih salah satu algoritma sorting seperti Bubble Sort, Selection Sort, Insertion Sort, atau Merge Sort. Pilih algoritma yang menurut Anda menarik untuk diimplementasikan dan dianalisis.
2. Tulis pseudocode untuk algoritma sorting yang Anda pilih. Pseudocode harus jelas dan mendekati bahasa pemrograman yang akan digunakan dalam implementasi nantinya.
3. Implementasikan algoritma sorting berdasarkan pseudocode yang telah Anda buat menggunakan bahasa pemrograman pilihan (misalnya, Python, Java, C++). Pastikan kode dapat menerima input berupa data yang akan diurutkan.
4. Hitung dan analisis kompleksitas waktu dari algoritma sorting yang diimplementasikan. Jelaskan bagaimana kompleksitas waktu berubah seiring dengan pertumbuhan ukuran data input.
5. Buatlah visualisasi sederhana yang menggambarkan proses sorting pada algoritma yang diimplementasikan. Anda dapat menggunakan diagram atau grafik untuk membantu memahami langkah-langkah algoritma. Gunakan contoh dengan ukuran input yang relatif kecil untuk memudahkan analisis Anda.
6. Tulis laporan yang mencakup deskripsi singkat algoritma yang diimplementasikan, pseudocode, analisis kompleksitas waktu, analisis efisiensi algoritma, dan hasil visualisasi. Sertakan kesimpulan mengenai kelebihan dan kelemahan algoritma sorting yang dipilih.