

## TUGAS INDIVIDU

Nama : Yudha Ananda Ramadhan

Npm : 22670136

Kelas : 3E Informatika

Soal:

1. Bagaimana [algorithma Dinamis](#) dalam menyelesaikan masalah ?
2. Bagaimana langkah atau cara kerja [algorithma Dinamis](#) tersebut ?
3. Bagaimana menentukan kompleksitas algorithmanya ?

Jawab:

1. Inti dari program dinamis menurut saya adalah membuang satu bagian kecil dari sebuah persoalan dalam setiap langkahnya, kemudian menyelesaikan persoalan yang lebih kecil tersebut dan menggunakan solusi hasil penyelesaian ini untuk ditambahkan kembali ke bagian persoalan dalam langkah berikutnya.
2. Berikut adalah langkah-langkah umum dalam pengembangan algoritma dinamis:

Karakteristikan Struktur Solusi Optimal:

- Identifikasi tahap-tahap (stage), variabel keputusan, dan status (state) yang terkait dengan masalah.
- Setiap tahap memiliki sejumlah status yang mungkin.
- Graf multistage (multistage graph) digunakan untuk menggambarkan hubungan antara tahap dan status.

Definisikan Nilai Solusi Optimal secara Rekursif:

- Hubungkan nilai optimal suatu tahap dengan tahap sebelumnya.
- Gunakan rekursi untuk menghitung nilai solusi optimal.

Hitung Nilai Solusi Optimal:

- Perhitungan dapat dilakukan secara maju (forward) atau mundur (backward).
- Hasil dari keputusan pada setiap tahap ditransformasikan ke tahap berikutnya.

Konstruksi Solusi Optimal:

- Dengan memanfaatkan hasil perhitungan, konstruksi solusi optimal dilakukan.

3. Berikut adalah langkah-langkah untuk menghitung kompleksitas algoritma:

-Tentukan Parameter (n): Identifikasi parameter (n) yang mengindikasikan ukuran input. Misalnya, pada algoritma pengurutan, (n) dapat mewakili jumlah elemen yang akan diurutkan.

Identifikasi Basic Operation:

- Temukan operasi dasar (basic operation) yang paling dominan dalam algoritma. Basic operation adalah operasi yang paling sering dieksekusi dan mempengaruhi kinerja algoritma. Contoh basic operation: perbandingan, penjumlahan, pengurangan, dll.

Tentukan Kasus Terburuk, Rata-rata, dan Terbaik: Analisis kompleksitas dilakukan dengan mempertimbangkan tiga kasus:

- Worst Case: Ketika algoritma memerlukan waktu terbanyak. Misalnya, pada algoritma pencarian linear, worst case terjadi ketika elemen yang dicari berada di posisi terakhir.

- Average Case: Rata-rata waktu eksekusi algoritma. Ini melibatkan distribusi input yang lebih realistis.

- Best Case: Waktu tercepat yang mungkin terjadi. Biasanya terjadi ketika algoritma memiliki kondisi khusus yang mempercepat eksekusi.

- Hitung Jumlah Basic Operation: Berdasarkan analisis di atas, tentukan berapa kali basic operation dieksekusi dalam algoritma. Ini akan membantu kita memahami bagaimana kompleksitas algoritma berubah seiring dengan ukuran masukan (n).