**TUGAS**

**Struktur Data**

Tugas ini disusun untuk memenuhi tugas Mata Kuliah Struktur Data

Dosen Pengampu : Muhammad Reksa Ariansyah, M.Kom.



Disusun Oleh

Fauzi Darmawan

22552011094

TIF RM – 22A Offline

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**SEKOLAH TINGGI TEKNOLOGY BANDUNG**

**2022**

Notasi O Besar atau Big-O Notation

Notasi O besar atau yang lazim disebut dengan Big-O Notation adalah sebuah cara atau metode untuk melakukan analisa terhadap sebuah algoritma pemrograman terhadap waktu eksekusi. Tentang seberapa efisien dan kompleksitas barisan kode dalam dimensi waktu.

Di dalam sebuah program komputer pada umumnya, kita sudah lazim dengan istilah masukan-proses-keluaran.

Beberapa Contoh Notasi O Besar

Mari kita melihat contoh notasi O besar dari fungsi, ekspresi dan operasi JavaScript yang sederhana.

## 1. O(1) — Constant Time

let myArray = [1, 5, 0, 6, 1, 9, 9, 2];  
function getFirst(input){ return input[0]; // selalu melakukan 1 langkah}let firstEl = getFirst(myArray);

Contoh diatas, terdapat sebuah fungsi untuk mengambil elemen pertama dari sebuah input array. Kita bisa melihat bahwa berapapun jumlah array yang diberikan kepada fungsi tersebut, dia akan selalu melakukan 1 hal, yaitu mengambil elemen pertama. Itu artinya **jumlah input yang diberikan tidak mempengaruhi waktu proses (**runtime**) dari algoritma tersebut**.

2. Array.pop()

pop() merupakan sebuah metode yang mengambil item terakhir dari array sehingga jumlah item yang ada di array akan berkurang satu.

3. Array.unshift()

unshift() adalah sebuah metode untuk menambahkan satu atau beberapa item ke bagian awal dari sebuah array. Contoh penggunaannya sebagai berikut.

## 4. O(log n) — Logarithmic Time

**Binary Search** adalah algoritma yang kita gunakan dalam mencari posisi nilai dari suatu array dengan cara ‘mengeliminasi’ setengah dari array input untuk mempercepat proses pencarian.

## 5. O(n) — Linear Time

let myArray = [1, 5, 0, 6, 1, 9, 9, 2];  
function getMax(input){  
 var max = 0; for (var i=0; i<input.length; i++){ if (max < input[i])  
 max = input[i]; }  
 return max;}let maxNumber = getMax(myArray);

## 6. O(n²) — Quadratic Time

let myArray = [1, 5, 0, 6, 1, 9, 9, 2];  
function sort(input){  
 var sortedArray = []; for (var i=0; i<input.length; i++){ // O(n) let min = input[i]; for (var j=i+1; i<input.length; i++){ // O(n) if (input[i] < input[j])  
 min = input[j]; } sortedArray.push(min); } return sortedArray;}let sortedArray = sort(myArray);

**Quadratic Time** adalah ketika runtime dari fungsi kita adalah sebesar n^2, dimana n adalah jumlah input dari fungsi tersebut. Hal tersebut bisa terjadi karena kita menjalankan **fungsi linear didalam fungsi linear** (n\*n).

Kesimpulan

Sebagai kesimpulan, notasi O besar atau Big-O Notation merupakan metode untuk menghitung kompleksitas dari potongan kode yang kita buat. Sehingga dapat menumbuhkan kesadaran kita untuk mencari alternatif yang lebih baik sebelum data semakin besar dan berdampak kepada performa aplikasi yang kita buat.

Notasi O besar inipun tidak hanya semata berlaku di bagian kode yang kita tulis, namun di database pun berlaku. Jadi proses pengambilan data di database dengan sintaksis SQL yang dapat dianggap sebagai proses perulangan akan sangat tidak efektif jika di bagian algoritma kode kita kembali menggunakan perulangan. Maka proses tersebut akan menjadi O(n2) karena akan terdapat perulangan didalam perulangan.

Begitu juga halnya jika kita mengambil data dengan tabel yang sudah diindeks. Secara otomatis notasinya akan berubah dari O(n) menjadi O(log n).