****

**ALGORITMA DAN STRUKTUR DATA**

INF1083

**LAPORAN PRAKTIKUM 4 : Analisis Algoritma (1)**

Oleh :

*Akhmad Qasim*

*2211102441237*

Teknik Informatika

Fakultas Sains & Teknologi

Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2023

**Laporan Praktikum 4:**

**Analisis Algoritma (1)**

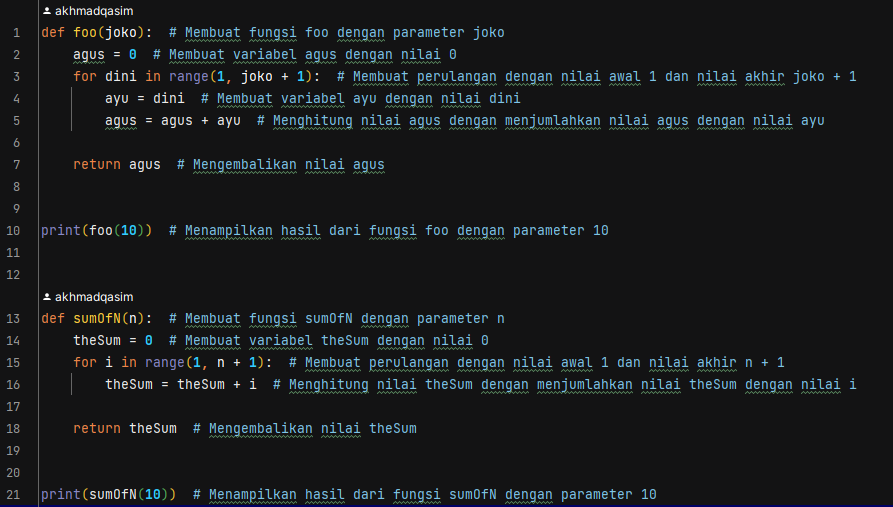
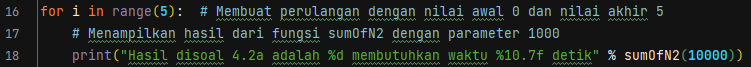
**Pokok Bahasan:**

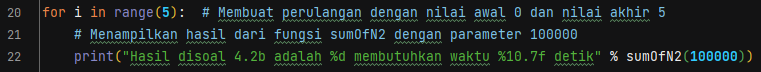
* Komparasi Algoritma
* Notasi Big O
* Algoritma *Checking Off*
* Algoritma *Sort & Compare*
* Algoritma *Count & Compare*

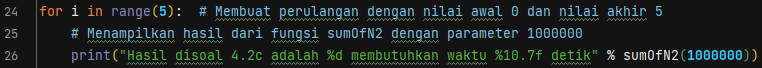
**Tujuan Pembelajaran:**

* Memahami cara menganalisis algoritma dengan komparasi hasil program.
* Memahami notasi Big O.
* Mengatasi masalah menggunakan algoritma *Checking Off, Sort & Compare, dan Count & Compare*.

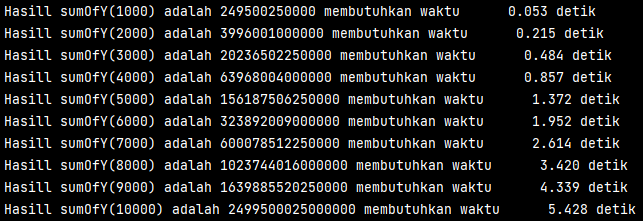
**Percobaan & Latihan:**

1. **Komparasi Algoritma**
   1. Berikan tampilan output kedua fungsi tersebut dan analisa!  
      
   2. Tentukan fungsi mana yang terbaik dan berikan alasannya!  
      Fungsi sumOfN() lebih baik dibandingkan fungsi foo() karena fungsi sumOfN() lebih mudah dibaca dan lebih mudah dipahami.
2. **Komparasi Algoritma**
   1. Jalankan perintah iterasi berikut dan analisa!  
        
      Text

      Description automatically generated
   2. Jalankan perintah iterasi berikut dan analisa!  
        
      Text

      Description automatically generated
   3. Jalankan perintah iterasi berikut dan analisa!  
        
      Text

      Description automatically generated
   4. Bandingkan hasil analisa ketiga iterasi tersebut dan berikan keterangan!  
      Terdapat perbedaan waktu yang sangat signifikan antara fungsi sumOfN2 dengan fungsi sumOfN. Fungsi sumOfN2 membutuhkan waktu yang lebih lama daripada fungsi sumOfN, hal ini dikarenakan jumlah perulangan yang lebih banyak pada fungsi sumOfN2.
3. **Komparasi Algoritma**
   1. Analisa perbandingan hasil iterasi dari fungsi diatas dengan hasil iterasi yang terdapat di Percobaan & Latihan 4.2a,b,c!  
      4.2 dengan 4.3 memiliki perbedaan dalam cara menghitung hasil iterasi. 4.2 menggunakan perulangan for untuk menghitung jumlah iterasi sedangkan 4.3 menggunakan rumus matematis untuk menghitung jumlah iterasi. Karena 4.3 menggunakan rumus matematis, maka waktu yang dibutuhkan untuk menghitung jumlah iterasi jauh lebih singkat dibandingkan dengan waktu yang dibutuhkan oleh 4.2. Karena itu, hasil iterasi pada kedua file tersebut akan sama, namun waktu yang dibutuhkan untuk menghitung iterasi pada 4.3 akan lebih singkat dibandingkan dengan 4.2.
   2. Buatlah perbandingan hasil waktu dari fungsi sumOfN3(1000000) dengan hasil waktu dari Percobaan & Latihan 4.2c, analisa dan tentukan fungsi mana yang terbaik berdasarkan waktu proses!  
      4.2c  
      Text

      Description automatically generated  
      sumOfN3(1000000)  
        
      Jelas bahwa fungsi sumOfN3 lebih cepat daripada fungsi sumOfN2. Hal ini dikarenakan fungsi sumOfN3 hanya melakukan 1 perhitungan. Sedangkan fungsi sumOfN2 melakukan perhitungan sebanyak n + 1 menggunakan perulangan for.
4. **Notasi Big O**
   1. Berikan tampilan output!  
      
   2. Buatlah grafik hasil fungsi diatas terhadap waktu (grafik silahkan menggunakan excel)!
   3. Termasuk notasi big O apakah algoritma diatas?  
      Notasi O(n^2)
5. **Algoritma Checking Off**
   1. Berikan tampilan output dan penjelasan tiap baris program!  
      Text

      Description automatically generated  
        
      Text

      Description automatically generated  
        
      Output:  
      A picture containing text, dark, orange, close

      Description automatically generated
6. **Algoritma Sort & Compare**
   1. Berikan tampilan output dan penjelasan tiap baris program!  
      A screenshot of a computer

      Description automatically generated with medium confidence  
        
      Output:  
      A picture containing text, orange, dark, close

      Description automatically generated
7. **Algoritma Count & Compare**
   1. Berikan tampilan output dan penjelasan tiap baris program!  
      Text

      Description automatically generated  
        
      Output:  
      

**Tugas Mandiri:**

O(n):  
A screenshot of a computer

Description automatically generated with low confidence  
  
O(n3):  
Text

Description automatically generated

**Kesimpulan:**

Analisis algoritma adalah proses mempelajari performa suatu algoritma dalam menyelesaikan sebuah masalah. Hal ini dilakukan dengan menganalisis waktu dan ruang yang dibutuhkan untuk menyelesaikan masalah dengan algoritma tersebut.

Notasi Big O adalah salah satu cara untuk mengukur kompleksitas waktu dari sebuah algoritma. Notasi Big O memberikan estimasi atas jumlah operasi yang diperlukan oleh sebuah algoritma dalam menyelesaikan sebuah masalah dalam rentang waktu tertentu. Dalam notasi Big O, waktu diukur sebagai fungsi dari ukuran input.

Anagram adalah sebuah istilah dalam bahasa Inggris yang merujuk pada sebuah kata atau frasa yang dibentuk dari huruf yang sama namun memiliki urutan yang berbeda. Contohnya adalah kata "debit card" dan "bad credit". Anagram dapat digunakan dalam pengkodean kata sandi atau dalam keamanan data, karena memiliki kemampuan untuk menyembunyikan arti dari sebuah pesan yang dikirim.

Kesimpulannya, analisis algoritma adalah proses untuk mempelajari performa suatu algoritma dalam menyelesaikan masalah, notasi Big O adalah cara untuk mengukur kompleksitas waktu dari sebuah algoritma, dan anagram adalah istilah yang merujuk pada kata atau frasa yang dibentuk dari huruf yang sama namun dengan urutan yang berbeda.