

INF218

Struktur Data & Algoritma

Pendahuluan

Alim Misbullah, S.Si., M.S.

Kontrak Kuliah

- Mata Kuliah : Struktur Data & Algoritma 3 SKS (2-1)
- Pengajar : Alim Misbullah, S.Si., M.S.
- Kontak : Group WA (**No Personal Chat**)
- Komting : Riska Adelia (STK)
: Abdul Hafidh (INF)

Kontrak Kuliah

- Mata kuliah SDA akan membahas tentang konsep-konsep yang digunakan untuk mempercepat kinerja sebuah program seperti linked list, queue, stack, sorting, searching sehingga waktu yang diperlukan untuk menyelesaikan sebuah masalah akan lebih cepat dari metode sederhana lainnya
- Pada mata kuliah ini, mahasiswa diharapkan sudah memiliki kemampuan dasar programming khususnya bahasa pemrograman C sehingga nantinya akan lebih mudah dalam melakukan implementasi terhadap konsep-konsep yang diajarkan.

Review...

- Sebuah program komputer adalah sekumpulan perintah untuk menyelesaikan masalah tertentu. Dalam hal ini, program komputer perlu untuk menyimpan data, membaca data dan melakukan perhitungan pada data tersebut.
- Struktur data merupakan sebuah istilah yang digunakan untuk menyimpan dan mengorganisir data. Sedangkan, algoritma adalah kumpulan langkah-langkah untuk menyelesaikan masalah tertentu
- Struktur Data dan Algoritma merupakan salah satu konsep yang harus dipelajari agar kita dapat menulis program komputer secara efisien dan optimal

Apa itu Struktur Data?

Struktur Data

- Struktur data adalah sebuah cara yang digunakan untuk mengorganisir data sehingga data tersebut dapat digunakan secara efisien (waktu dan tempat).
- Struktur data sangat erat kaitannya dengan penggunaan memori pada komputer sehingga data yang disimpan akan dengan mudah dapat diakses dan dilakukan manipulasi seperti penambahan, pencarian, penghapusan, pengeditan
- Struktur data **bukan** bahasa pemrograman seperti C, C++, Java, dll. SD merupakan kumpulan algoritma yang dapat diimplementasikan dalam berbagai bahasa pemrograman

Jenis Struktur Data

- Primitive Data Structure
 - Primitive Struktur Data adalah tipe data primitive seperti **int**, **float**, **double**, **char** dan **pointer** yang hanya dapat menampung satu nilai saja
 - `int a = 50; // variable a bertipe integer (-232 s.d 232-1), 4 bytes = 32 bits`
- Non-primitive Data Structure
 - Linear Data Structure
 - Data yang tersusun secara sequential (berurutan) dimana satu elemen terkoneksi hanya dengan satu element lainnya, seperti **array**, **linked list**, **stacks**, dan **queue**
 - `int arr[5] = {1}`
 - Non-linear Data Structure
 - Elemen yang terkoneksi ke banyak elemen lainnya, seperti **tree** dan **graph**

Jenis Struktur Data

- Static Data Structure
 - Jenis struktur data dimana ukuran memori akan dialokasikan pada saat kode atau program dikompilasi. Oleh karena itu, ukuran maksimum sudah pasti.
- Dynamic Data Structure
 - Jenis struktur data dimana ukuran memori akan dialokasikan pada saat kode atau program dijalankan. Oleh karena itu, ukuran maksimumnya sangat fleksibel

Operasi Pada Struktur Data

- Searching (Pencarian)
- Sorting (Pengurutan)
- Insertion (Penambahan)
- Updation (Pengeditan)
- Deletion (Penghapusan)

Keuntungan Struktur Data

- Efficiency (Efisiensi)
- Reusability (Digunakan berulang-ulang)
- Abstraction (Bias)

Apa itu Algoritma?

Algoritma

- Algoritma adalah sekumpulan instruksi (perintah) atau langkah-langkah yang tersusun dengan rapi untuk menyelesaikan sebuah masalah
- Kualitas sebuah Algoritma:
 - Input dan output dapat didefinisikan dengan tepat
 - Setiap langkah dalam sebuah algoritma harus jelas dan tidak ambigu
 - Algoritma yang digunakan adalah yang paling efektif diantara banyak solusi untuk menyelesaikan masalah
 - Sebuah algoritma seharusnya tidak memasukkan kode program. Bahkan, algoritma seharusnya ditulis untuk diimplementasikan dalam berbagai bahasa pemrograman

Contoh Algoritma

Algorithm to add two numbers entered by the user

```
Step 1: Start
Step 2: Declare variables num1, num2 and sum.
Step 3: Read values num1 and num2.
Step 4: Add num1 and num2 and assign the result to sum.
        sum←num1+num2
Step 5: Display sum
Step 6: Stop
```

<https://www.programiz.com/dsa/algorithm#add>

Contoh Algoritma

Find the largest number among three different numbers

```
Step 1: Start
Step 2: Declare variables a,b and c.
Step 3: Read variables a,b and c.
Step 4: If a > b
        If a > c
            Display a is the largest number.
        Else
            Display c is the largest number.
    Else
        If b > c
            Display b is the largest number.
        Else
            Display c is the greatest number.
Step 5: Stop
```

Contoh Algoritma Lainnya...

Algorithm to find all the roots of the quadratic equation

Algorithm to find the factorial

Algorithm to check prime number

Algorithm of Fibonacci series

Mengapa Perlu Belajar Algoritma?

Mengapa Perlu Algoritma?

- Waktu sangat berharga
 - Penyelesaian sebuah masalah (problem) di komputer akan lebih singkat jika memiliki algoritma yang efisien.
 - Contoh: Hitunglah penjumlahan deretan bilangan 10^{11}
 - Algoritma Sederhana

```
Initialize sum = 0
for every natural number n in range 1 to 1011 (inclusive):
    add n to sum
sum is your answer
```

- Code

```
int findSum() {
    int sum = 0;
    for (int v = 1; v <= 100000000000; v++) {
        sum += v;
    }
    return sum;
}
```

Butuh waktu setidaknya 16 menit

Mengapa Perlu Algoritma?

- Algoritma yang efisien

```
Sum = N * (N + 1) / 2
```

- Code

```
int sum(int N) {  
    return N * (N + 1) / 2;  
}
```

Hanya butuh milisecond

Mengapa Perlu Algoritma?

- Skalabilitas
 - Artinya, sebuah algoritma akan mampu bekerja dengan baik pada space yang lebih besar atau ukuran data yang lebih besar
 - Solusi pertama pada contoh sebelumnya disebut dengan **Linearly Scalable Algorithm** dan solusi kedua disebut dengan **Constant-Time Algorithm**
- Memori is Expensive
 - Contoh sederhana, kita dapat menyimpan informasi **umur (age)** dari sebuah entity tanpa perlu menyimpan tanggal lahir karena kita dapat menghitung tanggal lahir pada saat run time

Algoritma menjadi konsep dasar untuk melakukan optimisasi terhadap kode program sehingga menghemat waktu dan memori

Referensi

- <https://www.programiz.com/dsa>
- <https://www.javatpoint.com/data-structure-tutorial>
- https://www.tutorialspoint.com/data_structures_algorithms/index.htm