

Book Series
of
jago{coding}
Learn & Share {Your Code}

www.jagocoding.com

Metode Sequential Searching di Java Console

Oleh: Yudi Setiawan

Dalam ilmu pemrograman, melakukan pencarian Data merupakan hal yang paling sering dilakukan dalam perusahaan. Pencarian memiliki beberapa jenis metode yakni, Sequential Searching dan Binnary Searching. Pada tutorial ini, saya akan membahas tentang metode pencarian menggunakan metode Sequential Searching. ...

Dalam ilmu pemrograman, melakukan pencarian Data merupakan hal yang paling sering dilakukan dalam perusahaan. Pencarian memiliki beberapa jenis metode yakni, Sequential Searching dan Binnary Searching. Pada tutorial ini, saya akan membahas tentang metode pencarian menggunakan metode Sequential Searching. Dalam melakukan pencarian terhadap sekumpulan Data yang banyak ternyata proses pencarian akan membutuhkan waktu proses yang cukup lama apabila metode pencarian yang Anda gunakan tidak cocok atau sesuai dengan kondisi yang ada. Bisa Anda bayangkan bagaimana perusahaan sebesar Google bisa melakukan pencarian yang begitu cepat ketika Anda mengetik sesuatu hal pada mesin pencarinya dan itu tidak memerlukan waktu yang begitu lama.

“Apa itu Sequential Searching ?” Sequential Searching merupakan salah satu metode pencarian yang paling mudah digunakan dan sederhana dalam proses pencariannya dimana, proses pencariannya dilakukan dengan cara melakukan cek satu per satu terhadap data yang ada secara berurutan. Jadi, di dalam Sequential Searching data tidak perlu di urutkan namun, langsung dilakukan pengecekan. Untuk lebih jelasnya, langsung masuk ke contoh kasus yang ada. Misal, terdapat sejumlah Data seperti berikut.

Data : 12 15 9 1 20 25 -30

Data yang dicari : 20

Proses Sequential Searching :

- Pencarian dilakukan dari Data Pertama sampai Data Terakhir.
- Jika Data yang dicari berada di urutan yang paling jauh maka, Proses Sequential Searching akan memakan waktu yang cukup lama. Namun, sebaliknya jika Data yang dicari berada di urutan awal / dekat maka, proses Sequential Searching akan cepat.
- Bandingkan Data ke-1 dengan Data yang dicari $12 == 20 \rightarrow$ (False. Yang berarti bukan ini Data yang dicari)
- Bandingkan Data ke-2 dengan Data yang dicari $15 == 20 \rightarrow$ (False. Yang berarti bukan ini Data yang dicari)
- Bandingkan Data ke-3 dengan Data yang dicari $9 == 20 \rightarrow$ (False. Yang berarti bukan ini Data yang dicari)
- Bandingkan Data ke-4 dengan Data yang dicari $1 == 20 \rightarrow$ (False. Yang berarti bukan ini Data yang dicari)
- Bandingkan Data ke-5 dengan Data yang dicari $20 == 20 \rightarrow$ (True. Data yang dicari sudah ditemukan)
- Proses Pencarian berhenti.

Gimana mudahkan ? Sekarang coba lanjut lagi ke Contoh kasus kedua. Misal, terdapat sejumlah Data seperti berikut.

Data : 1 9 3 22 9 12 6 73

Data yang dicari : 9

Proses Sequential Searching :

- Pencarian dilakukan dari Data Pertama sampai Data Terakhir. Apabila terdapat Data yang sama atau duplikat maka, Data yang pertama kali ditemukanlah yang di cari.
- Bandingkan Data ke-1 dengan Data yang dicari $1 == 9 \rightarrow$ (False. Data belum ditemukan)
- Bandingkan Data ke-2 dengan Data yang dicari $9 == 9 \rightarrow$ (True. Data ditemukan)
- Proses Pencarian berhenti dan menampilkan bahwa Data ditemukan.

Berikut merupakan source code untuk program Sequential Searching.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.util.Random;

/**
 *
 * @author Yudi Setiawan
 *
 * Sequential Searching.
 *
 */

public class SequentialSearching
{
    public static void main(String[] args) throws IOException
    {
        //    Objek BufferedReader
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader dataIn = new BufferedReader(isr);

        //    Input jumlah Data
        System.out.print("Masukkan jumlah Data : ");
        int jlh_data = Integer.parseInt(dataIn.readLine());

        //    Array Data sebagai penampung nilai Data
        int[] data = new int[jlh_data];

        //    Pilih Metode Pengisian Data
        System.out.println("\nPENGISIAN DATA");
        System.out.println("1. Di input oleh user");
        System.out.print("2. Di input oleh program --> Pilihan [1 / 2]:");

        int isi = Integer.parseInt(dataIn.readLine());
        switch(isi)
        {
            case 1 : //    Pengisian Data dilakukan oleh si User
```

```

        System.out.println();
        for(int a = 0; a < jlh_data; a++)
        {
            System.out.print("Data ke-"+(a+1)+" : ");
            data[a] = Integer.parseInt(dataIn.readLine());
        }
        break;

        case 2 : // Pengisian Data dilakukan oleh program
secara random
            for(int a = 0; a < jlh_data; a++)
                data[a] = new Random().nextInt(201); //
Data di isi secara random dengan nilai tertinggi 200

            // Tampilkan Data yang di isi oleh program
            System.out.println("\nTampilkan Data");
            for(int a = 0; a < jlh_data; a++)
                System.out.println("Data ke-"+(a+1)+" :
"+data[a]);

            break;

        default : System.out.println("Pilihan tidak tersedia");
// Jika pilihan menu tidak sesuai
    }

    // Input Data yang dicari
    System.out.print("\nInput Data yang dicari : ");
    int cari = Integer.parseInt(dataIn.readLine());

    // Proses Sequential Searching
    boolean temu = false; // Variable indikator jika data
sudah ditemukan
    int urutan = 0; // Variable untuk mengetahui
urutan ke berapa data yang dicari
    for(int a = 0; a < jlh_data; a++)
    {
        System.out.print(data[a]+" == "+cari);
        if(data[a] == cari)
        {
            System.out.println(" --> Data ketemu");
            urutan = a+1; // Kenapa a+1 ? Karena a merupakan
posisi index dimana index dimulai dari 0 ...
            temu = true; // Sedangkan user menghitungnya dari 1.
            break;
        }

        else
            System.out.println(" --> Data belum ketemu");
    }
}

```

```
        if(temu == true)
            System.out.println("\nData ditemukan pada urutan
ke-"+(urutan));

        else
            System.out.println("\nData tidak ditemukan");
    }
}
```

Tentang Penulis



Yudi Setiawan

Saat ini aktif sebagai Mahasiswa di salah satu Universitas di kota Medan dengan mengambil bidang Fakultas Teknik dan Ilmu Komputer. Sangat senang dengan bahasa pemrograman Java dan Android.