

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

www.jagocoding.com

Bubble Sort dan Selection Sort di Java Console

Oleh: Yudi Setiawan

Dalam Perusahaan, tentu Data itu sangat penting. Bisa Anda bayangkan bahwa data &&&&– data dalam perusahaan itu sangat banyak. Kesalahan sedikit saja dalam proses Pengolahan Data dapat menyebabkan tidak kesesuaian data atau lebih parahnya lagi data tersebut hilang yang bisa disebabkan oleh program itu s...

Dalam Perusahaan, tentu Data itu sangat penting. Bisa Anda bayangkan bahwa data - data dalam perusahaan itu sangat banyak. Kesalahan sedikit saja dalam proses Pengolahan Data dapat menyebabkan tidak kesesuaian data atau lebih parahnya lagi data tersebut hilang yang bisa disebabkan oleh program itu sendiri atau bisa juga disebabkan dari faktor Human Error. Oleh karena itu, sebuah program yang baik ialah program yang mampu melakukan algoritma yang baik terhadap pengolahan data yang ada. Pada artikel ini, akan saya jelaskan bagaimana melakukan Pengurutan (Sorting) menggunakan metode Bubble Sort dan Selection Sort. Dalam artikel ini, saya akan membuatnya dalam bentuk bahasa pemrograman Java.

A) Bubble Sort

Bubble Sort merupakan metode pengurutan yang paling banyak digunakan di kalangan programmer dikarenakan penggunaannya yang simple dan sederhana. Namun, dibalik kesederhanaannya itu terdapat proses algoritma yang terlalu lama sehingga bisa dikatakan bahwa Metode Bubble Sort merupakan metode yang paling lambat dibanding dengan metode pengurutan yang lainnya. "Bagaimana sih Algoritma dari Bubble Sort tersebut?" Berikut akan saya jelaskan secara sederhana dengan contoh kasus seperti di bawah ini.

Data : 10 5 2 15 1

Proses Bubble Sort(Ascending):

Iterasi 1 :

10 5 2 15 1 → (10 bandingkan dengan 5)
5 10 2 15 1 → (10 tukar dengan 5. Bandingkan 10 dengan 2)
5 2 10 15 1 → (2 tukar dengan 10. Bandingkan 10 dengan 15)
5 2 10 15 1 → (Tidak ada pertukaran. Bandingkan 15 dengan 1)
5 2 10 1 15 → (15 tukar dengan 1)

Iterasi 2 :

5 2 10 1 15 → (5 bandingkan dengan 2)
2 5 10 1 15 → (5 tukar dengan 2. Bandingkan 5 dengan 10)
2 5 10 1 15 → (Tidak ada pertukaran. Bandingkan 10 dengan 1)
2 5 1 10 15 → (10 tukar dengan 1)

Iterasi 3 :

2 5 1 10 15 → (2 bandingkan dengan 5)

2 5 1 10 15 → (Tidak ada pertukaran. Bandingkan 5 dengan 1)
2 1 5 10 15 → (5 tukar dengan 1)

Iterasi 4 :

2 1 5 10 15 → (2 bandingkan dengan 1)
1 2 5 10 15 → (2 tukar dengan 1)

Iterasi 5 :

1 2 5 10 15 → (1 bandingkan dengan 2)

Maka, Data diatas setelah di sorting ialah sebagai berikut :

1 2 5 10 15

Penjelasan Algoritma Bubble Sort:

- Dalam Bubble Sort. Jumlah Iterasi sebesar banyaknya Data. Datas jumlah datanya ialah 5 maka, jumlah iterasinya ialah 5. Selain itu, setiap iterasi terdapat proses yang jumlahnya ialah sebesar banyaknya Data. Datas jumlah datanya ialah 5 maka, jumlah proses setiap iterasinya ialah 5. Dan untuk iterasi berikutnya harus dikurang 1.
- Dalam Bubble Sort, Walaupun data sudah terurut seperti pada kasus diatas, data sudah terurut pada iterasi ke-4. Namun, proses sorting tetap jalan sampai jumlah iterasinya terpenuhi.
- Proses Pertukaran Datanya dimulai dari data pertama dibandingkan dengan data kedua atau bisa digambarkan dengan $Data[n] \leq Data[n+1]$. Lakukan langkah ini sampai berada pada Data Terakhir.

Dari proses algoritma diatas dapat saya simpulkan bahwa Metode Bubble Sort merupakan metode sorting yang paling lambat dibandingkan dengan metode sorting yang lain karena, terdapat proses perbandingan yang terlalu banyak walaupun Data sudah terurut. Dan berikut ialah coding untuk program Bubble Sort.

```
import java.util.Scanner;

/**
 *
 * @author Yudi Setiawan
 *
 * Bubble Sort
 *
 */

public class BubbleSort
{
    public static void main(String[] args)
    {
        //    Buat Objek Scanner
        Scanner scan = new Scanner(System.in);

        //    Input jumlah Data
        System.out.print("Masukkan jumlah Data : ");    int jlh_data =
```

```

scan.nextInt();

//    Input nilai tiap Data
int[] data = new int[jlh_data];          //    Array untuk menampung
nilai tiap Data
System.out.println();
for(int a = 0; a < jlh_data; a++)
{
    System.out.print("Nilai Data ke-" + (a+1) + " : ");
    data[a] = scan.nextInt();
}

//    Tampilkan Data Sebelum di Sorting
System.out.println("\nData Sebelum di Sorting");
for(int a = 0; a < jlh_data; a++)
    System.out.print(data[a] + " ");

//    Proses Bubble Sort
System.out.println("\nProses Bubble Sort");
for(int a = 0; a < jlh_data; a++)
{
    System.out.println("Iterasi ke-" + (a+1) + " :");
    for(int b = 0; b < jlh_data; b++)
        System.out.print(data[b] + " ");

    System.out.println("    Bandingkan " + data[0] + " dengan
"+data[1]);
    for(int b = 0; b < jlh_data-1; b++)
    {
        String pesan = "    Tidak ada pertukaran";
        if(data[b] > data[b+1])
        {
            //    proses pertukaran nilai Data
            pesan = "    Data " + data[b] + " ditukar dengan
"+data[b+1];
            int temp = data[b];          //    Variable Sebagai
pihak ketiga
            data[b] = data[b+1];
            data[b+1] = temp;

        }

        if(b < jlh_data - (a+1))
        {
            for(int c = 0; c < jlh_data; c++)
                System.out.print(data[c] + " ");

            System.out.println(pesan);
        }
    }
}

```

```

        System.out.println("\n");
    }

    //    Tampilkan Data Setelah di Sorting
    System.out.print("Data Setelah di Sorting : ");
    for(int a = 0; a < jlh_data; a++)
        System.out.print(data[a]+" ");

    }
}

```

B) Selection Sort

Selection Sort merupakan salah satu metode pengurutan yang memiliki algoritma yang cukup gampang dalam penulisan coding-nya. Dibanding Bubble Sort, Selection Sort jelas lebih baik dari segi kecepatan proses pengurutannya. Karena, inti dari algoritma Selection Sort ialah mencari nilai yang paling kecil (Jika Ascending) atau nilai yang paling besar (Jika Descending) di urutan Data berikutnya. Untuk lebih jelasnya saya beri contoh kasus seperti berikut ini.

Data : 13 9 15 2 3 1

Proses Selection Sort (Ascending)

Iterasi 1 :

13 9 15 2 3 1 → (Apakah 13 nilai yang paling kecil?)

1 9 15 2 3 13 → (Tidak. 13 ditukar dengan 1)

Iterasi 2 :

1 9 15 2 3 13 → (Apakah 9 nilai yang paling kecil?)

1 2 15 9 3 13 → (Tidak. 9 ditukar dengan 2)

Iterasi 3 :

1 2 15 9 3 13 → (Apakah 15 nilai yang paling kecil?)

1 2 3 9 15 13 → (Tidak. 15 ditukar dengan 3)

Iterasi 4 :

1 2 3 9 15 13 → (Apakah 9 nilai yang paling kecil?)

1 2 3 9 15 13 → (Iya)

Iterasi 5 :

1 2 3 9 15 13 → (Apakah 15 nilai yang paling kecil?)

1 2 3 9 13 15 → (Tidak. 15 ditukar dengan 13)

Data Setelah di sorting ialah sebagai berikut :

Data : 1 2 3 9 13 15

Penjelasan Algoritma Selection Sort:

- Jumlah Iterasi untuk Selection Sort ialah berjumlah sebesar Jumlah Data - 1. Untuk kasus diatas, Jumlah Datanya ialah 6. Maka, jumlah Iterasinya ialah sebesar $6 - 1 = 5$.
- Proses pertukaran Data dimulai dari Data Pertama sampai Data Terakhir dengan cara

membandingkan Data ke-n dan cari nilai yang paling kecil di sisi kanan nilai n.

- Keterangan bahwa nilai Data yang sudah di tukar(nilai yang paling kecil) tidak akan dibandingkan lagi untuk proses iterasi berikutnya. Berikut ilustrasi lengkapnya untuk kasus diatas.

Warna Merah : Data yang akan ditukar atau cek apakah Data tersebut sudah pas di posisinya.

Warna Cyan : Cari nilai yang paling kecil atau Data belum terurut.

Warna Hijau : Data yang ditukar atau Tidak ada pertukaran Data.

Warna Abu-abu : Data yang sudah terurut atau sudah pas di posisinya.



Dan berikut ialah source code untuk metode Selection Sort.

```
import java.util.Scanner;

/**
 *
 * @author Yudi Setiawan
 *
 * Selection Sort
 *
 */

public class SelectionSort
{
    public static void main(String[] args)
    {
        //    Buat Objek Scanner
        Scanner scan = new Scanner(System.in);

        //    Input jumlah Data
        System.out.print("Masukkan jumlah Data : ");    int jlh_data =
scan.nextInt();

        //    Input nilai tiap Data
        int[] data = new int[jlh_data];                //    Array untuk nilai
tiap Data
        System.out.println();
        for(int x = 0; x < jlh_data; x++)
        {
            System.out.print("Input nilai Data ke-" + (x+1) + " : ");
            data[x] = scan.nextInt();
        }

        //    Tampilkan Data Sebelum di sorting
        System.out.println();
        System.out.print("Data Sebelum di Sorting : ");
```

```

for(int x = 0; x < jlh_data; x++)
    System.out.print(data[x]+" ");

//    Proses Selection Sort
System.out.println("\n\nProses Selection Sort");
for(int x = 0; x < jlh_data-1; x++)
{
    System.out.println("Iterasi ke-"+(x+1)+" : ");
    for(int y = 0; y < jlh_data; y++)
        System.out.print(data[y]+" ");

    System.out.println("    Apakah Data "+data[x]+" sudah benar
pada urutannya?");

    boolean tukar = false;
    int index = 0;
    int min = data[x];
    String pesan = "    Tidak Ada Pertukaran";
    for(int y = x+1; y < jlh_data; y++)
    {
        if(min > data[y])
        {
            tukar = true;
            index = y;
            min = data[y];
        }
    }

    if(tukar == true)
    {
        //    Pertukaran Data
        pesan = "    Data "+data[x]+" ditukar dengan Data
"+data[index];
        int temp = data[x];
        data[x] = data[index];
        data[index] = temp;
    }

    for(int y = 0; y < jlh_data; y++)
        System.out.print(data[y]+" ");

    System.out.println(pesan+"\n");
}

//    Tampilkan Data Setelah di Sorting
System.out.print("Data Setelah di sorting : ");
for(int x = 0; x < jlh_data; x++)
    System.out.print(data[x]+" ");
}

```

}

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.