

Perbandingan Metode Selection Sort dan Insertion Sort dalam Pengurutan Data Menggunakan Bahasa Program Java

Endang Sunandar

*Jurusan Sistem Komputer Universitas Raharja
Jalan Jendral Sudirman No 40 Modernland Cikokol Tangerang
endang.sunandar@raharja.info*

ABSTRACT

There are various kinds of sorting data methods that we know of which are Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Shell Sort, and Heap Sort methods. All of these methods have their respective strengths and weaknesses, the use of which is determined based on needs. Each method has a different algorithm, where the difference in this algorithm affects the execution time. In this paper the authors make a comparison of the 2 methods of sorting data, namely the Sort Sort and Insertion Sort methods, with consideration that the two methods are concise algorithms and have almost the same algorithm pattern. , using the same number and data model. The purpose of this comparison is to provide an overview of the two methods, which method has faster execution time, whether the Selection sort method or the Insertion Sort method.

Keywords: *Selection Sort, Insertion Sort, Algorithm, Execution Time*

ABSTRAK

Terdapat berbagai macam metode pengurutan data yang kita ketahui diantaranya adalah metode Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Shell Sort, dan Heap Sort. Seluruh metode tersebut memiliki kelebihan dan kelemahan masing-masing, yang penggunaannya ditentukan berdasarkan kebutuhan. Setiap metode memiliki algoritma yang berbeda, dimana perbedaan algoritma ini mempengaruhi waktu eksekusi. Dalam tulisan ini penulis melakukan perbandingan terhadap 2 metode pengurutan data yaitu metode Selection Sort dan Insertion Sort, dengan pertimbangan adalah bahwa ke-dua metode ini algoritmanya ringkas dan memiliki pola algoritma yang hampir sama. Perbandingan yang dilakukan adalah dengan melihat jumlah baris program dan waktu eksekusinya, dengan menggunakan jumlah dan model data yang sama. Adapun tujuan dari melakukan perbandingan ini adalah untuk memberikan gambaran terhadap 2 metode tersebut, metode mana yang memiliki waktu eksekusi lebih cepat, apakah metode Selection sort atau metode Insertion Sort.

Kata kunci: *Selection Sort, Insertion Sort, Algoritma, Waktu Eksekusi*

1. PENDAHULUAN

Berbagai pola algoritma pengurutan data yang ada memiliki kelebihan dan kekurangan masing-masing. Penggunaan masing-masing algoritma tersebut sangat tergantung dari kebutuhan dan kondisi yang ada. Umumnya pemilihan suatu algoritma pengurutan data didasarkan pada kesederhanaan / keringkasannya dan kecepatan proses eksekusinya.

Beberapa metode pengurutan data yang ada diantaranya adalah Selection Sort, Insertion Sort, Bubble Sort, Merge Sort, Quick Sort, Shell Sort, Heap Sort, dan Radix Sort. Dalam penelitian ini penulis mengambil secara random terhadap 2 buah metode pengurutan, yaitu metode Selection Sort dan Insertion. Pertimbangannya adalah bahwa ke-dua metode ini sangat sederhana, ringkas, dan mudah untuk dipahaminya, sehingga relatif banyak digunakan.

Hasil yang diharapkan dari penelitian ini adalah untuk membantu mempermudah analisis terhadap pemilihan metode untuk pengurutan data, apakah Selection Sort atau Insertion Sort, tentunya disesuaikan dengan tingkat kebutuhan di lapangan.

Penelitian sebelumnya yang pernah ada adalah "Implementasi Metode Selection Sort untuk Menentukan Nilai Prestasi Siswa Kelas 3 dan Kelas 4 SD Negeri 107 Seluma"[1]. Dalam penelitian tersebut digambarkan tentang bagaimana kumpulan data nilai siswa diurutkan secara Descending (dari besar ke kecil) dengan metode Selection Sort menggunakan bahasa program Visual Basic.Net 2010 dan database SQL Server 2018r2. Output dari hasil penelitian ini adalah menyajikan daftar nilai siswa mulai dengan urutan terbesar hingga terkecil. Penelitian selanjutnya yang berjudul "Perbandingan Algoritma Insertion Sort dengan Merge Sort Pada Bahasa Pemrograman C dan Java" menyampaikan bahwa metode Merge Sort lebih cepat dibandingkan dengan metode Insertion Sort menggunakan dua jenis dataset random dan descending sebanyak lima puluh ribu data[2]. Sedangkan dalam penelitian yang berjudul "Algoritma Pengurutan Data (Sorting) dengan Metode Insertion Sort dan Selection Sort" menyatakan bahwa algoritma Selection Sort menggunakan prinsip pertukaran elemen data, sedangkan algoritma Insertion Sort menggunakan prinsip geser dan sisip elemen data[3].

2. METODE / PERANCANGAN PENELITIAN

Pada penelitian ini, peneliti menggunakan 2 buah metode pengurutan data yaitu **Selection Sort** dan **Insertion Sort** pada suatu deretan data, dan data akan diurutkan secara Ascending (dari kecil ke besar) bergerak dari kiri ke kanan. Untuk implementasinya menggunakan bahasa pemrograman Java.

2.1. Selection Sort

Selection sort adalah suatu metode pengurutan data dengan cara memilih suatu data pada urutan tertentu, kemudian membandingkannya dengan data-data lainnya mulai dari posisi [posisi data+1] sampai dgn data pada posisi ke-n, untuk mencari data terkecil pada rentang posisi tersebut. Jika data terkecil ditemukan, maka pindahkan data terkecil tersebut ke posisi [posisi data], dan data yang semula berada di posisi [posisi data] dipindahkan ke posisi dimana data terkecil tadi ditemukan. Demikian seterusnya hingga data terakhir.

Ilustrasi:

Index	0	1	2	3	4	5	6	7
Urutan	1	2	3	4	5	6	7	8
Data	G	I	B	J	K	R	U	V

Algoritmanya:

1. Carilah data terkecil dengan cara membandingkan data yang berada di urutan ke-1 (G) dengan data-data yang berada di urutan ke-2 (I) sampai urutan ke-8 (V), jika data terkecil tersebut ditemukan di lokasi antara urutan ke-2 sampai urutan ke-8, maka pindahkan data terkecil tersebut ke posisi urutan ke-1, dan data yang semula berada di posisi ke-1 dipindahkan ke posisi dimana data terkecil tadi ditemukan.
2. Carilah data terkecil dengan cara membandingkan data yang berada di urutan ke-2 (I) dengan data-data yang berada di urutan ke-3 (B) sampai urutan ke-8 (V), jika data terkecil tersebut ditemukan di lokasi antara urutan ke-3 sampai urutan ke-8, maka pindahkan data terkecil tersebut ke posisi urutan ke-2, dan data yang semula berada di posisi ke-2 dipindahkan ke posisi dimana data terkecil tadi ditemukan.
3. Demikian seterusnya sampai dengan data di urutan ke-7.

2.2. Insertion Sort

Insertion sort adalah suatu metode pengurutan data dengan cara menyimpan data ke suatu variabel sementara, kemudian dibandingkan dengan data-data lainnya yang ada disebelah kiri posisi data tersebut. Demikian seterusnya hingga data terakhir.

Ilustrasi:

Index	0	1	2	3	4	5	6	7
Urutan	1	2	3	4	5	6	7	8
Data	G	I	B	J	K	R	U	V

Algoritmanya:

1. Simpan data yang ada di urutan ke-2 (I) ke dalam suatu variabel buffer, lalu bandingkan data tersebut (I) dengan data yang berada di urutan ke-1 (G). Jika data I lebih kecil dari data G, maka data G digeser pindah ke posisi urutan ke-2, lalu data I (yang semula berada di variabel buffer) dipindahkan ke posisi di urutan ke-1. Tetapi jika data I lebih besar dari data G maka data I yang semula berada di variabel buffer dikembalikan lagi ke posisi urutan ke-2.
2. Simpan data yang ada di urutan ke-3 (B) ke dalam suatu variabel buffer, lalu bandingkan data tersebut (B) dengan data-data yang ada di sebelah kirinya (data di urutan ke-2 (I) dan di urutan ke-1 (G)). Jika data yang ada di variabel buffer (B) < dari data yang ada di urutan ke-2 (I) maka data I bergeser ke posisi urutan ke-3. Lalu bandingkan data B dengan data yang ada di urutan ke-1 (G). Jika data B < dari data yang ada di urutan ke-1 (G) maka data G bergeser ke posisi urutan ke-2, lalu data B dipindahkan ke posisi urutan ke-1. Tetapi jika data B > dari data G, maka data B dipindahkan ke posisi urutan ke-2.
3. Demikian seterusnya sampai dengan data di urutan ke-8.

3. HASIL DAN PEMBAHASAN

Dalam pembangunan penelitian ini sumber data yang digunakan adalah beberapa data angka, dimana listing programnya adalah seperti berikut ini:

Program Java Selection Sort:

```
* @author Ir.Endang Sunandar.MKom
```

```
*/
```

```
//selection sort ascending - dgn menempatkan nilai lebih kecil di kiri
```

```
import java.util.Date;
```

```
public class selectionSORTNomer {
```

```
public static int terkecil,posisi;
public static void main(String args[]){
    int [] data={ 64,60,32,99,28,16,9,11,35,12,14,1,15,7};
    int posisi;
    System.out.println("Metode SELECTION SORT");
    System.out.println("Data Awal : 64,60,32,99,28,16,9,11,35,12,14,1,15,7");
    System.out.println("-----");
    //awal looping jumlah data
    Date date = new Date();
    System.out.println("Tanggal skrg : "+date.toString());
    System.out.println("-----");
    for (int i=0;i<=12;i++) {
        terkecil=data[i];
        int buffer=data[i];
        posisi=i;
        //awal looping ke kanan
        for (int j=i+1;j<=13;j++){
            if (data[j] < terkecil) {
                terkecil=data[j];
                posisi=j;
            }
        } // batas akhir looping ke kanan
        data[i]=terkecil;
        data[posisi]=buffer;
        //-----
        for (int k=0;k<=13;k++)
        {
            System.out.print(data[k]+" ");
        }
        System.out.println(" <----- Urutan data ke : "+(i+1));
        //-----
    } // batas akhir looping jumlah data
    System.out.println("-----");
    System.out.println("Tanggal skrg : "+date.toString());
    System.out.println("-----");
} }
```

- **Hasil output metode Selection Sort adalah berikut ini :**

Metode INSERTION SORT

Data awal : 64,60,32,99,28,16,9,11,35,12,14,1,15,7

Tanggal skrg : Thu Jul 04 22:57:56 ICT 2019

60, 64, 32, 99, 28, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 1
32, 60, 64, 99, 28, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 2
32, 60, 64, 99, 28, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 3

28, 32, 60, 64, 99, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 4
16, 28, 32, 60, 64, 99, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 5
9, 16, 28, 32, 60, 64, 99, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 6
9, 11, 16, 28, 32, 60, 64, 99, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 7
9, 11, 16, 28, 32, 35, 60, 64, 99, 12, 14, 1, 15, 7, <----- Urutan data ke : 8
9, 11, 12, 16, 28, 32, 35, 60, 64, 99, 14, 1, 15, 7, <----- Urutan data ke : 9
9, 11, 12, 14, 16, 28, 32, 35, 60, 64, 99, 1, 15, 7, <----- Urutan data ke : 10
1, 9, 11, 12, 14, 16, 28, 32, 35, 60, 64, 99, 15, 7, <----- Urutan data ke : 11
1, 9, 11, 12, 14, 15, 16, 28, 32, 35, 60, 64, 99, 7, <----- Urutan data ke : 12
1, 7, 9, 11, 12, 14, 15, 16, 28, 32, 35, 60, 64, 99, <----- Urutan data ke : 13

Tanggal skrg : Thu Jul 04 22:57:56 ICT 2019

Program Java Insertion Sort :

```
/*@author Ir.Endang Sunandar.MKom
*
/
import java.util.Date;
public class InsertionSort {
    public static int buffer;
    public static int data[]={64,60,32,99,28,16,9,11,35,12,14,1,15,7};
    public static void main(String args[]){
        System.out.println("Metode INSERTION SORT");
        System.out.println("Data awal : 64,60,32,99,28,16,9,11,35,12,14,1,15,7");
        System.out.println("-----");
        Date date = new Date();
        System.out.println("Tanggal skrg : "+date.toString());
        System.out.println("-----");
        for (int i=1; i<14 ; i++){
            buffer=data[i];
            for (int j =i-1;j>=0; j--){
                int geser=j;
                if (buffer <data[j]){
                    data[j+1]=data[j];
                    data[j]=buffer;
                }
            } // akhir looping j
            for (int k=0;k<=13;k++){
                System.out.print(data[k]+" , ");
            } System.out.println(" <----- Urutan data ke : "+(i));
        } // akhir looping i
        System.out.println("-----");
        System.out.println("Tanggal skrg : "+date.toString());
        System.out.println("-----");
    }
}
```

- **Hasil output metode Insertion Sort adalah berikut ini :**

Metode INSERTION SORT
Data awal : 64,60,32,99,28,16,9,11,35,12,14,1,15,7

Tanggal skrg : Sat Jul 06 09:07:36 ICT 2019

60, 64, 32, 99, 28, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 1
32, 60, 64, 99, 28, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 2
32, 60, 64, 99, 28, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 3
28, 32, 60, 64, 99, 16, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 4
16, 28, 32, 60, 64, 99, 9, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 5
9, 16, 28, 32, 60, 64, 99, 11, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 6
9, 11, 16, 28, 32, 60, 64, 99, 35, 12, 14, 1, 15, 7, <----- Urutan data ke : 7
9, 11, 16, 28, 32, 35, 60, 64, 99, 12, 14, 1, 15, 7, <----- Urutan data ke : 8
9, 11, 12, 16, 28, 32, 35, 60, 64, 99, 14, 1, 15, 7, <----- Urutan data ke : 9
9, 11, 12, 14, 16, 28, 32, 35, 60, 64, 99, 1, 15, 7, <----- Urutan data ke : 10
1, 9, 11, 12, 14, 16, 28, 32, 35, 60, 64, 99, 15, 7, <----- Urutan data ke : 11
1, 9, 11, 12, 14, 15, 16, 28, 32, 35, 60, 64, 99, 7, <----- Urutan data ke : 12
1, 7, 9, 11, 12, 14, 15, 16, 28, 32, 35, 60, 64, 99, <----- Urutan data ke : 13

Tanggal skrg : Sat Jul 06 09:07:36 ICT 2019

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan pada perbandingan 2 metode Selection Sort dan Insertion Sort ini maka dapat diambil beberapa kesimpulan sebagai berikut :

Sepintas tidak ada perbedaan waktu mencolok dari penggunaan metode Selection Sort dan Insertion Sort ini, dengan jumlah data relatif sedikit (13 data) hanya membutuhkan waktu sekitar 1 detik untuk mengeksekusinya. Untuk penelitian berikutnya dapat menggunakan data dengan ukuran yang lebih banyak, untuk mengetahui perbedaan waktu eksekusi secara signifikan.

Algoritma Selection Sort terlihat lebih sederhana karena hanya membandingkan suatu data dengan data-data lain yang berada pada rentang posisi tertentu, dan jika ditemukan data terkecilnya(pada rentang posisi tersebut) langsung ditukarkan dengan data yang dibandingkan tadi.

Algoritma Insertion Sort sedikit lebih kompleks, karena memerlukan suatu variabel penampung data sementara, untuk menampung data yang akan dibandingkan dengan data-data sebelumnya, kemudian ada proses geser (bukan pindah) yang mengimplementasikan proses secara sequential, sehingga waktu proses relatif lebih lama dibandingkan dengan proses random (pindah langsung, pada metode Selection Sort).

Saran lebih lanjut adalah mengimplementasikan 2 buah metode ini dalam suatu aplikasi sistem tertentu, untuk mengetahui dan mendapatkan sisi manfaat yang lebih besar.

DAFTAR PUSTAKA

- [1] Benardo, Mesterjon, Julita Leni Natalia, [2015], *Implementasi Metode Selection Sort untuk Menentukan Nilai Prestasi Siswa Kelas 3 dan Kelas 4 SD Negeri 07 Seluma*”, Jl. Meranti Raya No 32 kota Bengkulu, Prodi Teknik Informatika Fakultas Ilmu Komputer Universitas Dehasen.

- [2] Primartha Rifkie [2015], "Perbandingan Algoritma Insertion Sort dengan Merge Sort pada Bahasa Pemrograman C dan Java", Jl. Srijaya Negara Bukit Lama Palembang, Jurusan Informatika Fakultas Ilmu Komputer Universitas Sriwijaya Palembang.
- [3] Retnoningsih Endang, [2018], "Algoritma Pengurutan Data (Sorting) dengan Metode Insertion Sort dan Selection Sort", Jl. Siliwangi No 6 Bekasi, Prodi Sistem Informasi STMIK Bina Insani.