LAPORAN PRAKTIKUM STRUKTUR DATA

Modul ke	: 5
Judul Praktikum	: Sorting
Hari dan Tanggal Pelaksanaan	: Senin, 25 April 2022
Tempat Pelaksanaan	: Lab Desain
Dosen Pengampu Praktikum	: Khoirul Umam, S.Pd, M.Kom
Nama Mahasiswa Pelaksana	: Andri Wijaksono
NIM Pelaksana	: 362155401206

A. Tugas Pendahuluan

Tuliskan hasil pengerjaan Tugas Pendahuluan pada bagian ini.

B. Kegiatan Praktikum

Cantumkan apa saja yang dilakukan pada latihan-latihan praktikum, *source code* yang dipakai, *screen shot* hasil eksekusi kode, dan jawaban dari pertanyaan-pertanyaan yang muncul pada tiap kegiatan latihan.

Gambarkan flowchart untuk masing-masing algoritma bubble sort, insertion sort, dan selection sort untuk pengurutan data secara ascending!

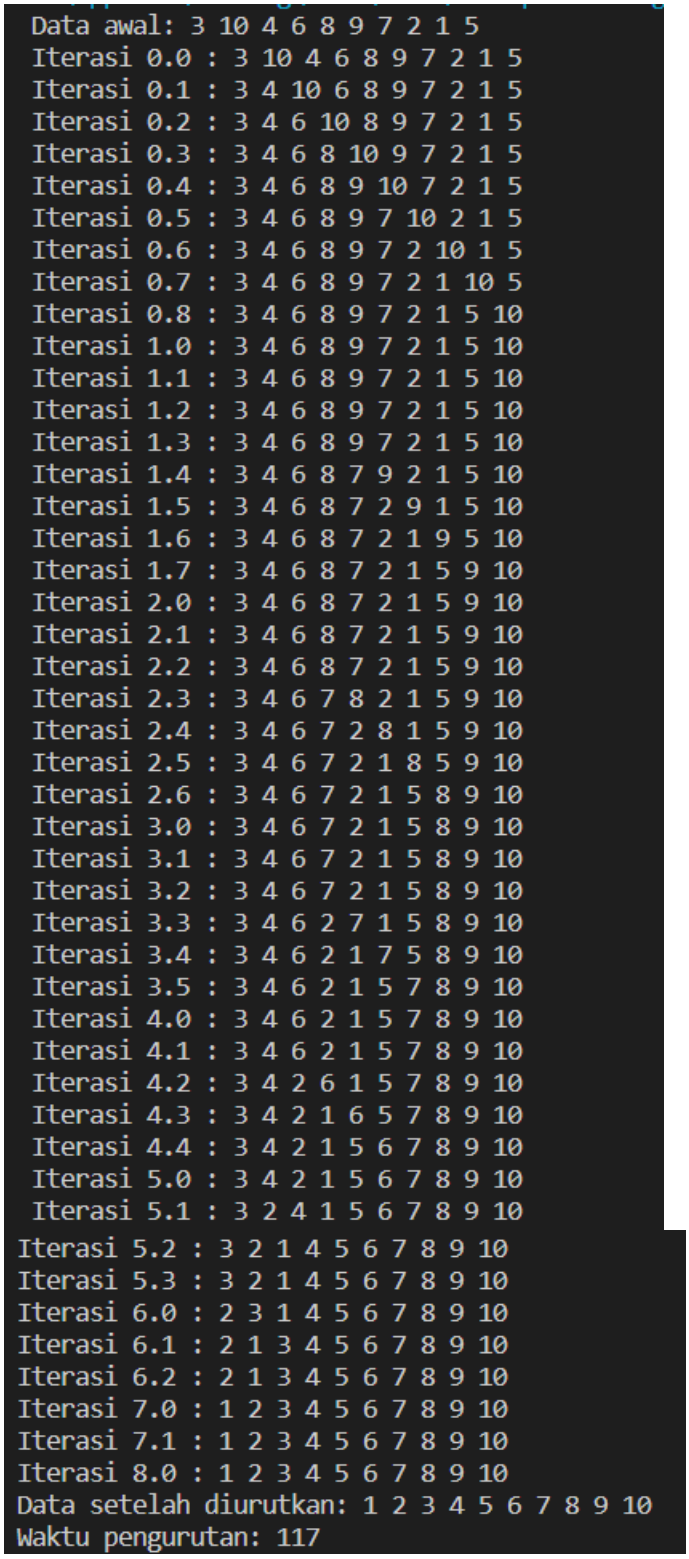
Latihan 1: Ascending bubble sort

1. Buat file dengan nama BubbleSort.java kemudian tuliskan kode berikut:

```
import java.lang.Comparable;
public class BubbleSort {
    public static <T extends Comparable<? super T>> void sort(T[] arr) {
        int i, j;
        T temp;

        for (i = 0; i < arr.length - 1; i++) {
    for (j = 0; j < arr.length - i - 1; j++) {
        if (arr[j].compareTo(arr[j + 1]) > 0) {
            temp = arr[j + 1];
            arr[j + 1] = arr[j];
            arr[j] = temp;
            }
        // log tahapan sorting
    System.out.print("Iterasi " + i + "." + j + " : ");
    tampil(arr);
    }
    }
}
    public static <T> void tampil(T data[]) {
    for (T objek : data) {
        System.out.print(objek + " ");
    }
```

```
        System.out.println();
    }
    public static void main(String[] args) {
    Integer data[] = { 3, 10, 4, 6, 8, 9, 7, 2, 1, 5 };
        System.out.print("Data awal: ");
            tampil(data);
            long awal = System.currentTimeMillis();
            sort(data);
            long selisihWaktu = System.currentTimeMillis() - awal;
            System.out.print("Data setelah diurutkan: ");
            tampil(data);
                System.out.println("Waktu pengurutan: " + selisihWaktu);
    }
}
```



3. Berapa kali iterasi yang dilakukan hingga didapatkan data dalam kondisi berurutan?
Jawab: 9 kali Literasi

4. Coba ubah himpunan data yang akan diurutkan menjadi 10, 1, 2, 3, 4, 5, 6, 7, 8, 9. Berapa jumlah iterasi yang dibutuhkan sampai didapatkan data terurut?
Jawab: 9 kali Literasi

```
Data awal: 10 1 2 3 4 5 6 7 8 9
Iterasi 0.0 : 1 10 2 3 4 5 6 7 8 9
Iterasi 0.1 : 1 2 10 3 4 5 6 7 8 9
Iterasi 0.2 : 1 2 3 10 4 5 6 7 8 9
Iterasi 0.3 : 1 2 3 4 10 5 6 7 8 9
Iterasi 0.4 : 1 2 3 4 5 10 6 7 8 9
Iterasi 0.5 : 1 2 3 4 5 6 10 7 8 9
Iterasi 0.6 : 1 2 3 4 5 6 7 10 8 9
Iterasi 0.7 : 1 2 3 4 5 6 7 8 10 9
Iterasi 0.8 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.4 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.5 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.6 : 1 2 3 4 5 6 7 8 9 10
Iterasi 1.7 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.4 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.5 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.6 : 1 2 3 4 5 6 7 8 9 10
Iterasi 3.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 3.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 3.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 3.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 3.4 : 1 2 3 4 5 6 7 8 9 10
Iterasi 3.5 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.4 : 1 2 3 4 5 6 7 8 9 10
Iterasi 5.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 5.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 5.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 5.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 6.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 6.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 6.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 7.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 7.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 8.0 : 1 2 3 4 5 6 7 8 9 10
Data setelah diurutkan: 1 2 3 4 5 6 7 8 9 10
Waktu pengurutan: 104
PS C:\Users\ASUS\Documents\Struktur Data>
```

Latihan 2: Ascending bubble sort dengan flag

1. Modifikasi method **sort**() pada class **BubbleSort** di atas menjadi seperti berikut:

```
import java.lang.Comparable;
public class BubbleSort {
    public static <T extends Comparable<? super T>> void sort(T[] arr) {
        int i = 0, j;
        T temp;
        Boolean didSwap = true; // flag
    while (i < arr.length - 1 && didSwap) {
    didSwap = false;

        for (i = 0; i < arr.length - 1; i++) {
    for (j = 0; j < arr.length - i - 1; j++) {
        if (arr[j].compareTo(arr[j + 1]) > 0) {
            temp = arr[j + 1];
            arr[j + 1] = arr[j];
            arr[j] = temp;
            didSwap = true;
            }
        // log tahapan sorting
    System.out.print("Iterasi " + i + "." + j + " : ");
    tampil(arr);
    }
    i++;
    }
    }
    }
    public static <T> void tampil(T data[]) {
    for (T objek : data) {
        System.out.print(objek + " ");
    }
        System.out.println();
    }
    public static void main(String[] args) {
    Integer data[] = { 10, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
        System.out.print("Data awal: ");
            tampil(data);
            long awal = System.currentTimeMillis();
            sort(data);
            long selisihWaktu = System.currentTimeMillis() - awal;
            System.out.print("Data setelah diurutkan: ");
            tampil(data);
                System.out.println("Waktu pengurutan: " + selisihWaktu);
    }
}
```

2. Gunakan kembali himpunan data 3, 10, 4, 6, 8, 9, 7, 2, 1, 5 untuk data yang akan diurutkan.

```
Data awal: 3 10 4 6 8 9 7 2 1 5
Iterasi 0.0 : 3 10 4 6 8 9 7 2 1 5
Iterasi 0.1 : 3 4 10 6 8 9 7 2 1 5
Iterasi 0.2 : 3 4 6 10 8 9 7 2 1 5
Iterasi 0.3 : 3 4 6 8 10 9 7 2 1 5
Iterasi 0.4 : 3 4 6 8 9 10 7 2 1 5
Iterasi 0.5 : 3 4 6 8 9 7 10 2 1 5
Iterasi 0.6 : 3 4 6 8 9 7 2 10 1 5
Iterasi 0.7 : 3 4 6 8 9 7 2 1 10 5
Iterasi 0.8 : 3 4 6 8 9 7 2 1 5 10
Iterasi 2.0 : 3 4 6 8 9 7 2 1 5 10
Iterasi 2.1 : 3 4 6 8 9 7 2 1 5 10
Iterasi 2.2 : 3 4 6 8 9 7 2 1 5 10
Iterasi 2.3 : 3 4 6 8 9 7 2 1 5 10
Iterasi 2.4 : 3 4 6 8 7 9 2 1 5 10
Iterasi 2.5 : 3 4 6 8 7 2 9 1 5 10
Iterasi 2.6 : 3 4 6 8 7 2 1 9 5 10
Iterasi 4.0 : 3 4 6 8 7 2 1 9 5 10
Iterasi 4.1 : 3 4 6 8 7 2 1 9 5 10
Iterasi 4.2 : 3 4 6 8 7 2 1 9 5 10
Iterasi 4.3 : 3 4 6 7 8 2 1 9 5 10
Iterasi 4.4 : 3 4 6 7 2 8 1 9 5 10
Iterasi 6.0 : 3 4 6 7 2 8 1 9 5 10
Iterasi 6.1 : 3 4 6 7 2 8 1 9 5 10
Iterasi 6.2 : 3 4 6 7 2 8 1 9 5 10
Iterasi 8.0 : 3 4 6 7 2 8 1 9 5 10
Data setelah diurutkan: 3 4 6 7 2 8 1 9 5 10
Waktu pengurutan: 72
PS C:\Users\ASUS\Documents\Struktur Data>
```

4. Berapa kali iterasi yang dibutuhkan untuk mendapatkan data-data dalam kondisi terurut? Apakah ada perbedaan dengan latihan sebelumnya untuk himpunan data yang sama?

Jawab: Terdapat 5 iterasi, perbedaannya adalah pada penggunaan BubbleSort dengan flag yang digunakan untuk menentukan apakah terjadi operasi swapping selama proses bubble up berlangsung. Sehingga waktu iterasi yang digunakan untuk mendapatkan sebuah data berurutan akan lebih sedikit.

5. Coba ubah kembali himpunan data yang akan diurutkan menjadi 10, 1, 2, 3, 4, 5, 6, 7, 8, 9. Berapa jumlah iterasi yang dibutuhkan sampai didapatkan data terurut? Apakah ada perbedaan dengan latihan sebelumnya untuk himpunan data yang sama?
Jawab: Pada jumlah iterasi yang dibutuhkan 5. Perbedaannya terdapat pada penggunaan BubbleSort dengan flag yang digunakan untuk menentukan hasil iterasi menjadi lebih sedikit atau simple.

```
Data awal: 10 1 2 3 4 5 6 7 8 9
Iterasi 0.0 : 1 10 2 3 4 5 6 7 8 9
Iterasi 0.1 : 1 2 10 3 4 5 6 7 8 9
Iterasi 0.2 : 1 2 3 10 4 5 6 7 8 9
Iterasi 0.3 : 1 2 3 4 10 5 6 7 8 9
Iterasi 0.4 : 1 2 3 4 5 10 6 7 8 9
Iterasi 0.5 : 1 2 3 4 5 6 10 7 8 9
Iterasi 0.6 : 1 2 3 4 5 6 7 10 8 9
Iterasi 0.7 : 1 2 3 4 5 6 7 8 10 9
Iterasi 0.8 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.4 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.5 : 1 2 3 4 5 6 7 8 9 10
Iterasi 2.6 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.3 : 1 2 3 4 5 6 7 8 9 10
Iterasi 4.4 : 1 2 3 4 5 6 7 8 9 10
Iterasi 6.0 : 1 2 3 4 5 6 7 8 9 10
Iterasi 6.1 : 1 2 3 4 5 6 7 8 9 10
Iterasi 6.2 : 1 2 3 4 5 6 7 8 9 10
Iterasi 8.0 : 1 2 3 4 5 6 7 8 9 10
Data setelah diurutkan: 1 2 3 4 5 6 7 8 9 10
Waktu pengurutan: 61
PS C:\Users\ASUS\Documents\Struktur Data>
```

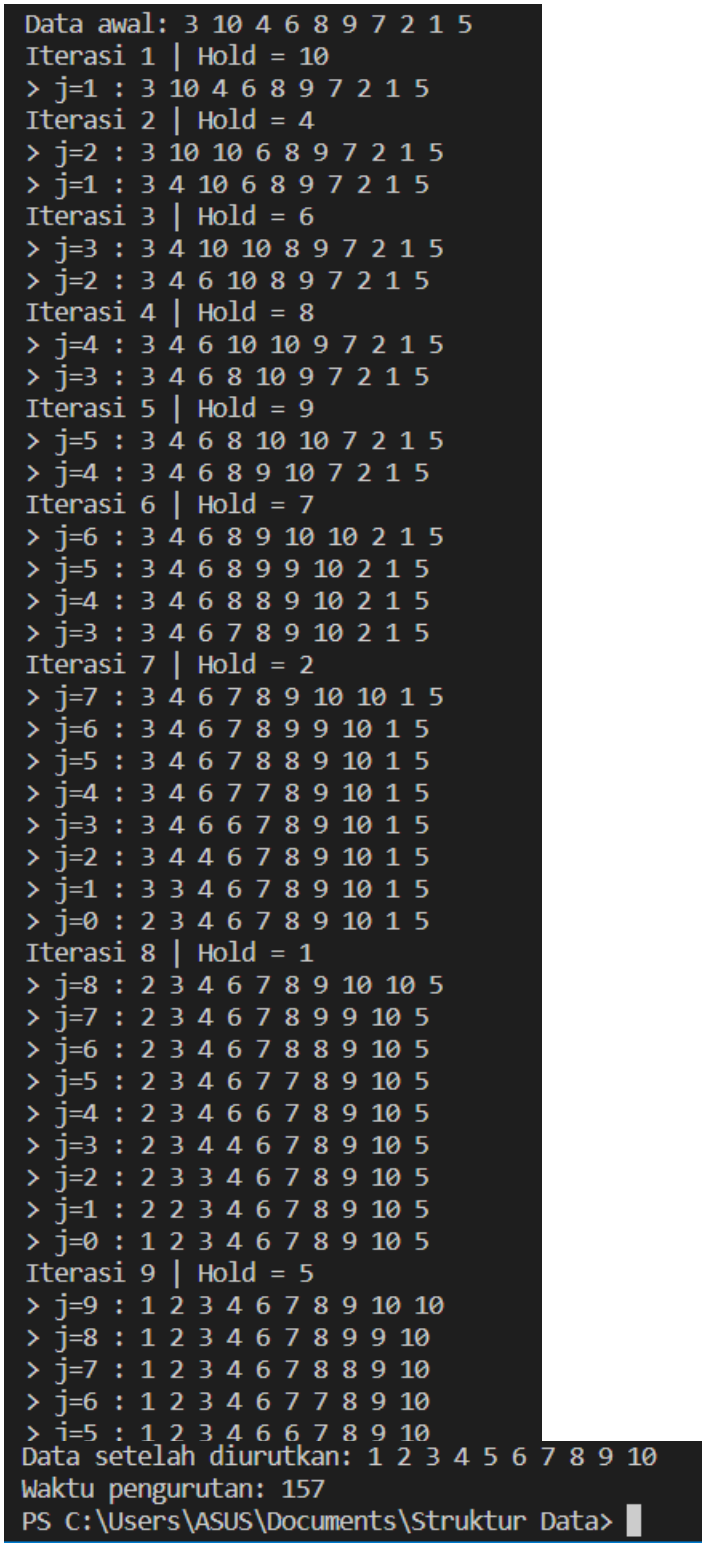
6. Apakah adanya flag dapat menghemat jumlah iterasi yang dibutuhkan? Jelaskan!
Jawab: Iya, karena flag melakukan pertukaran sebuah data melalui proses Bubble Up sehingga jumlah iterasi yang dihasilkan akan lebih simple.

Latihan 3: Ascending insertion sort

1. Buat file bernama **InsertionSort.java** kemudian tuliskan kode berikut:

```
import java.lang.Comparable;
public class InsertionSort {
public static <T extends Comparable<? super T>> void sort(T[] arr) {
int i, j;
T hold;
for (i = 1; i < arr.length; i++) {
```

```
hold = arr[i];
j = i;
System.out.println("Iterasi " + i + " | Hold = " + hold);
while (j > 0 && hold.compareTo(arr[j - 1]) < 0) {
arr[j] = arr[j - 1];
// log tahapan sorting
System.out.print("> j=" + j + " : ");
tampil(arr);
j--;
}
arr[j] = hold;
// log tahapan sorting
System.out.print("> j=" + j + " : ");
tampil(arr);
}
}
public static <T> void tampil(T data[]) {
for (T objek : data) {
System.out.print(objek + " ");
}
System.out.println();
}
public static void main(String[] args) {
Integer data[] = { 3, 10, 4, 6, 8, 9, 7, 2, 1, 5 };
System.out.print("Data awal: ");
tampil(data);
long awal = System.currentTimeMillis();
sort(data);
long selisihWaktu = System.currentTimeMillis() - awal;
System.out.print("Data setelah diurutkan: ");
tampil(data);
System.out.println("Waktu pengurutan: " + selisihWaktu);
}
}
```



3. Berapa kali iterasi yang dibutuhkan untuk mendapatkan data terurut?
Jawab: 9 kali literasi
4. Berapa total proses swapping data yang terjadi pada seluruh iterasi?
Jawab: 9 kaliswapping

5. Apa yang membedakan proses pengurutan pada algoritma insertion sort ini dengan algoritma bubble sort pada latihan sebelumnya? Jelaskan!
Jawab: Perbedaannya terdapat pada proses membandingkan urutannya. Pada BubbleSort pengurutannya dengan membandingkan urutannya. Sedangkan pada InsertionSort pengurutannya dengan cara menyisipkan data pada posisi dimana nilainya lebih besar dari sebelah kiri.

Latihan 4: Ascending selection sort

1. Buat file baru dengan nama SelectionSort.java kemudian tuliskan kode berikut:

```
import java.lang.Comparable;
public class SelectionSort {
    public static <T extends Comparable<? super T>> void sort(T[] arr) {
    T temp;
    for (int i = 0; i < arr.length; i++) {
    int minPos = i;
    System.out.println("Iterasi " + i + " | Minpos = " + minPos);
    for (int j = i + 1; j < arr.length; j++) {
    if (arr[j].compareTo(arr[minPos]) < 0) {
    minPos = j;
    }
    System.out.println("> Cek j=" + j + " : Minpos = " + minPos);
    }
    temp = arr[i];
    arr[i] = arr[minPos];
    arr[minPos] = temp;
    System.out.print("Swap posisi " + i + "-" + minPos + " : ");
    tampil(arr);
    }
    }
    public static <T> void tampil(T data[]) {
    for (T objek : data) {
    System.out.print(objek + " ");
    }
    System.out.println();
    }
    public static void main(String[] args) {
    Integer data[] = { 3, 10, 4, 6, 8, 9, 7, 2, 1, 5 };
    System.out.print("Data awal: ");
    tampil(data);
    long awal = System.currentTimeMillis();
    sort(data);
    long selisihWaktu = System.currentTimeMillis() - awal;
    System.out.print("Data setelah diurutkan: ");
    tampil(data);
    System.out.println("Waktu pengurutan: " + selisihWaktu);
    }
}
```

```
Data awal: 3 10 4 6 8 9 7 2 1 5
Iterasi 0 | Minpos = 0
> Cek j=1 : Minpos = 0
> Cek j=2 : Minpos = 0
> Cek j=3 : Minpos = 0
> Cek j=4 : Minpos = 0
> Cek j=5 : Minpos = 0
> Cek j=6 : Minpos = 0
> Cek j=7 : Minpos = 7
> Cek j=8 : Minpos = 8
> Cek j=9 : Minpos = 8
Swap posisi 0-8 : 1 10 4 6 8 9 7 2 3 5
Iterasi 1 | Minpos = 1
> Cek j=2 : Minpos = 2
> Cek j=3 : Minpos = 2
> Cek j=4 : Minpos = 2
> Cek j=5 : Minpos = 2
> Cek j=6 : Minpos = 2
> Cek j=7 : Minpos = 7
> Cek j=8 : Minpos = 7
> Cek j=9 : Minpos = 7
Swap posisi 1-7 : 1 2 4 6 8 9 7 10 3 5
Iterasi 2 | Minpos = 2
> Cek j=3 : Minpos = 2
> Cek j=4 : Minpos = 2
> Cek j=5 : Minpos = 2
> Cek j=6 : Minpos = 2
> Cek j=7 : Minpos = 2
> Cek j=8 : Minpos = 8
> Cek j=9 : Minpos = 8
Swap posisi 2-8 : 1 2 3 6 8 9 7 10 4 5
Iterasi 3 | Minpos = 3
> Cek j=4 : Minpos = 3
> Cek j=5 : Minpos = 3
> Cek j=6 : Minpos = 3
> Cek j=7 : Minpos = 3
> Cek j=8 : Minpos = 8
> Cek j=9 : Minpos = 8
Swap posisi 3-8 : 1 2 3 4 8 9 7 10 6 5
Iterasi 4 | Minpos = 4
> Cek j=5 : Minpos = 4
> Cek j=6 : Minpos = 6
> Cek j=7 : Minpos = 6
> Cek j=8 : Minpos = 8
> Cek j=9 : Minpos = 9
Swap posisi 7-9 : 1 2 3 4 5 6 7 8 9 10
Iterasi 8 | Minpos = 8
> Cek j=9 : Minpos = 8
Swap posisi 8-8 : 1 2 3 4 5 6 7 8 9 10
Iterasi 9 | Minpos = 9
Swap posisi 9-9 : 1 2 3 4 5 6 7 8 9 10
Data setelah diurutkan: 1 2 3 4 5 6 7 8 9 10
Waktu pengurutan: 94
PS C:\Users\ASUS\Documents\Struktur Data>
```

3. Berapa banyak proses swapping yang dibutuhkan untuk mendapatkan data terurut pada kasus di atas?
Jawab: 9 kali swapping
4. Berapa kali pencarian posisi nilai minimum pada contoh kasus di atas terjadi?
Jawab: 4 kali
5. Apa yang membedakan proses pengurutan pada algoritma selection sort ini dengan algoritma bubble sort dan insertion sort pada latihan sebelumnya? Jelaskan!
Jawab: Pengurutan pada proses SelectionSort ini dilakukan dengan cara membandingkan data dan mengurutkannya mulai dari data yang memiliki nilai yang terkecil hingga ke nilai yang terbesar. Sedangkan 2 proses yang lain diurutkan dengan data yang berada didekatnya maupun yang berada disebelah kiri dan kanan.

C. Tugas Praktikum

Tuliskan dan jabarkan hasil pengerjaan Tugas Praktikum yang tertera di dalam modul lengkap dengan *source code* yang digunakan.

1. Tambahkan method descendingSort() ke dalam class BubbleSort, InsertionSort, maupun SelectionSort agar pengurutan menggunakan metode-metode tersebut dapat menghasilkan urutan secara menurun (descending).
Jawab:

- Bubblesort

```
import java.lang.Comparable;

public class BubbleSort {
    public static <T extends Comparable<? super T>> void sort(T[] arr)
{
        int  i = 0, j;
        T temp;
        Boolean didSwap = true;

        while (i < arr.length - 1 && didSwap) {
            didSwap = false;


        for (i = 0; i < arr.length - 1; i++) {
            for (j = 0; j < arr.length - i - 1; j++) {
                if (arr[j].compareTo(arr[j + 1]) < 0) {
                    temp = arr[j + 1];
                    arr[j + 1] = arr[j];
                    arr[j] = temp;

                    didSwap = true;

                }

```

```
                System.out.print("Iterasi " + i + "." + j + " : ");
                tampil(arr);
            }

            i++;
        }
    }
}

    public static <T> void tampil(T data[]) {
        for (T objek : data) {
            System.out.print(objek + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Integer data[] = {  3, 10, 4, 6, 8, 9, 7, 2, 1, 5};

        System.out.print("Data awal: ");
        tampil(data);

        long awal = System.currentTimeMillis();
        sort(data);
        long selisihWaktu = System.currentTimeMillis() - awal;

        System.out.print("Data setelah diurutkan: ");
        tampil(data);
        System.out.println("Waktu pengurutan: " + selisihWaktu);
    }
}
```

```
Data awal: 3 10 4 6 8 9 7 2 1 5
Iterasi 0.0 : 10 3 4 6 8 9 7 2 1 5
Iterasi 2.1 : 10 6 4 8 9 7 3 2 5 1
Iterasi 2.2 : 10 6 8 4 9 7 3 2 5 1
Iterasi 2.3 : 10 6 8 9 4 7 3 2 5 1
Iterasi 2.4 : 10 6 8 9 7 4 3 2 5 1
Iterasi 2.5 : 10 6 8 9 7 4 3 2 5 1
Iterasi 2.6 : 10 6 8 9 7 4 3 2 5 1
Iterasi 4.0 : 10 6 8 9 7 4 3 2 5 1
Iterasi 4.1 : 10 8 6 9 7 4 3 2 5 1
Iterasi 4.2 : 10 8 9 6 7 4 3 2 5 1
Iterasi 4.3 : 10 8 9 7 6 4 3 2 5 1
Iterasi 4.4 : 10 8 9 7 6 4 3 2 5 1
Iterasi 6.0 : 10 8 9 7 6 4 3 2 5 1
Iterasi 6.1 : 10 9 8 7 6 4 3 2 5 1
Iterasi 6.2 : 10 9 8 7 6 4 3 2 5 1
Iterasi 8.0 : 10 9 8 7 6 4 3 2 5 1
Data setelah diurutkan: 10 9 8 7 6 4 3 2 5 1
Waktu pengurutan: 148
PS D:\Perkuliahan\pratikum5>
```

- Insertionsort

```
import java.lang.Comparable;

```

```java
public class InsertionSort {
    public static <T extends Comparable<? super T>> void sort(T[] arr)
{
        int i, j;
        T hold;

        for (i = 1; i < arr.length; i++) {
            hold = arr[i];
            j = i;

            System.out.println("Iterasi " + i + " | Hold = " + hold);

            while (j > 0 && hold.compareTo(arr[j - 1]) > 0) {
                arr[j] = arr[j - 1];

                System.out.print("> j=" + j + " : ");
                tampil(arr);

                j--;
            }

            arr[j] = hold;

            System.out.print("> j=" + j + " : ");
            tampil(arr);
        }
    }

    public static <T> void tampil(T data[]) {
        for (T objek : data) {
            System.out.print(objek + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Integer data[] = { 3, 10, 4, 6, 8, 9, 7, 2, 1, 5 };

        System.out.print("Data awal: ");
        tampil(data);

        long awal = System.currentTimeMillis();
        sort(data);
        long selisihWaktu = System.currentTimeMillis() - awal;

        System.out.print("Data setelah diurutkan: ");
        tampil(data);
        System.out.println("Waktu pengurutan: " + selisihWaktu);
```

```
    }
}
```

```
Data awal: 3 10 4 6 8 9 7 2 1 5
Iterasi 1 | Hold = 10
> j=1 : 3 3 4 6 8 9 7 2 1 5
> j=0 : 10 3 4 6 8 9 7 2 1 5
Iterasi 2 | Hold = 4
> j=2 : 10 3 3 6 8 9 7 2 1 5
> j=1 : 10 4 3 6 8 9 7 2 1 5
Iterasi 3 | Hold = 6
> j=3 : 10 4 3 3 8 9 7 2 1 5
> j=2 : 10 4 4 3 8 9 7 2 1 5
> j=1 : 10 6 4 3 8 9 7 2 1 5
Iterasi 4 | Hold = 8
> j=4 : 10 6 4 3 3 9 7 2 1 5
> j=3 : 10 6 4 4 3 9 7 2 1 5
> j=2 : 10 6 6 4 3 9 7 2 1 5
> j=1 : 10 8 6 4 3 9 7 2 1 5
Iterasi 5 | Hold = 9
> j=5 : 10 8 6 4 3 3 7 2 1 5
> j=4 : 10 8 6 4 4 3 7 2 1 5
> j=3 : 10 8 6 6 4 3 7 2 1 5
> j=2 : 10 8 8 6 4 3 7 2 1 5
> j=1 : 10 9 8 6 4 3 7 2 1 5
Iterasi 6 | Hold = 7
> j=6 : 10 9 8 6 4 3 3 2 1 5
> j=5 : 10 9 8 6 4 4 3 2 1 5
> j=4 : 10 9 8 6 6 4 3 2 1 5
> j=3 : 10 9 8 7 6 4 3 2 1 5
Iterasi 7 | Hold = 2
> j=7 : 10 9 8 7 6 4 3 2 1 5
Iterasi 8 | Hold = 1
> j=8 : 10 9 8 7 6 4 3 2 1 5
Iterasi 9 | Hold = 5
> j=9 : 10 9 8 7 6 4 3 2 1 1
> j=8 : 10 9 8 7 6 4 3 2 2 1
> j=7 : 10 9 8 7 6 4 3 3 2 1
> j=6 : 10 9 8 7 6 4 4 3 2 1
> j=5 : 10 9 8 7 6 5 4 3 2 1
Data setelah diurutkan: 10 9 8 7 6 5 4 3 2 1
Waktu pengurutan: 63
PS D:\Perkuliahan\pratikum5>
```

- Selectionsort

```
import java.lang.Comparable;

public class SelectionSort {
    public static <T extends Comparable<? super T>> void sort(T[] arr)
{
        T temp;

        for (int i = 0; i < arr.length; i++) {
            int minPos = i;

            System.out.println("Iterasi " + i + " | Minpos = " +
minPos);

            for (int j = i + 1; j < arr.length; j++) {
                if (arr[j].compareTo(arr[minPos]) > 0) {
                    minPos = j;
                }
```

```
                System.out.println("> Cek j=" + j + " : Minpos = " +
minPos);
            }

            temp = arr[i];
            arr[i] = arr[minPos];
            arr[minPos] = temp;

            System.out.print("Swap posisi " + i + "-" + minPos + " :
");
            tampil(arr);
        }
    }

    public static <T> void tampil(T data[]) {
        for (T objek : data) {
            System.out.print(objek + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Integer data[] = { 3, 10, 4, 6, 8, 9, 7, 2, 1, 5 };

        System.out.print("Data awal: ");
        tampil(data);

        long awal = System.currentTimeMillis();
        sort(data);
        long selisihWaktu = System.currentTimeMillis() - awal;

        System.out.print("Data setelah diurutkan: ");
        tampil(data);
        System.out.println("Waktu pengurutan: " + selisihWaktu);
    }
}
```

```
Data awal: 3 10 4 6 8 9 7 2 1 5
Iterasi 0 | Minpos = 0
> Cek j=1 : Minpos = 1
> Cek j=2 : Minpos = 1
> Cek j=3 : Minpos = 1
> Cek j=4 : Minpos = 1
> Cek j=5 : Minpos = 1
> Cek j=6 : Minpos = 1
> Cek j=7 : Minpos = 1
> Cek j=8 : Minpos = 1
> Cek j=9 : Minpos = 1
Swap posisi 0-1 : 10 3 4 6 8 9 7 2 1 5
Iterasi 1 | Minpos = 1
> Cek j=2 : Minpos = 2
> Cek j=3 : Minpos = 3
> Cek j=4 : Minpos = 4
> Cek j=5 : Minpos = 5
> Cek j=6 : Minpos = 5
> Cek j=7 : Minpos = 5
> Cek j=8 : Minpos = 5
> Cek j=9 : Minpos = 5
Swap posisi 1-5 : 10 9 4 6 8 3 7 2 1 5
Iterasi 2 | Minpos = 2
> Cek j=3 : Minpos = 3
> Cek j=4 : Minpos = 4
> Cek j=5 : Minpos = 4
> Cek j=6 : Minpos = 4
> Cek j=7 : Minpos = 4
> Cek j=8 : Minpos = 4
> Cek j=9 : Minpos = 4
Swap posisi 2-4 : 10 9 8 6 4 3 7 2 1 5
Iterasi 3 | Minpos = 3
> Cek j=4 : Minpos = 3
> Cek j=5 : Minpos = 3
> Cek j=6 : Minpos = 6
> Cek j=7 : Minpos = 6
> Cek j=8 : Minpos = 6
```

```
> Cek j=9 : Minpos = 6
Swap posisi 3-6 : 10 9 8 7 4 3 6 2 1 5
Iterasi 4 | Minpos = 4
> Cek j=5 : Minpos = 4
> Cek j=6 : Minpos = 6
> Cek j=7 : Minpos = 6
> Cek j=8 : Minpos = 6
> Cek j=9 : Minpos = 6
Swap posisi 4-6 : 10 9 8 7 6 3 4 2 1 5
Iterasi 5 | Minpos = 5
> Cek j=6 : Minpos = 6
> Cek j=7 : Minpos = 6
> Cek j=8 : Minpos = 6
> Cek j=9 : Minpos = 9
Swap posisi 5-9 : 10 9 8 7 6 5 4 2 1 3
Iterasi 6 | Minpos = 6
> Cek j=7 : Minpos = 6
> Cek j=8 : Minpos = 6
> Cek j=9 : Minpos = 6
Swap posisi 6-6 : 10 9 8 7 6 5 4 2 1 3
Iterasi 7 | Minpos = 7
> Cek j=8 : Minpos = 7
> Cek j=9 : Minpos = 9
Swap posisi 7-9 : 10 9 8 7 6 5 4 3 1 2
Iterasi 8 | Minpos = 8
> Cek j=9 : Minpos = 9
Swap posisi 8-9 : 10 9 8 7 6 5 4 3 2 1
Iterasi 9 | Minpos = 9
Swap posisi 9-9 : 10 9 8 7 6 5 4 3 2 1
Data setelah diurutkan: 10 9 8 7 6 5 4 3 2 1
Waktu pengurutan: 76
PS C:\Users\ASUS\Documents\Struktur Data>
```

2. Lakukan pengurutan pada array of objects dari class Mahasiswa yang pernah dibuat pada modul sebelumnya namun tidak dengan menggunakan Java native sorting method, melainkan dengan menggunakan user-defined method yang mengimplementasikan salah satu metode sorting pada modul ini.

Jawab: