

**LAPORAN PRAKTIKUM**  
**Algoritma dan Struktur Data**

**PRAKTIKUM 4 W5**  
**Sorting**



**Oleh:**  
**JESSICA AMELIA**  
**2341760185**  
**SIB 1A/16**

**PROGRAM STUDI SISTEM INFORMASI BISNIS**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2023/2024**

## Jobsheet 5

### SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

#### Percobaan 1

##### Mengurutkan Data Angka

1. Jika jumlah elemen array 5, ada berapa iterasi yang dilakukan pada masing-masing jenis sorting (untuk outer loop dan inner loop)?

**Jawab :**

- Bubble Sort
    - Outer loop : 5 iterasi
    - Inner loop :  
Iterasi pertama, inner loop berjalan 4 kali iterasi (karena elemen terakhir sudah di tempatnya)  
Pada iterasi kedua, inner loop berjalan 3 kali iterasi.  
Pada iterasi ketiga, inner loop berjalan 2 kali iterasi.  
Pada iterasi keempat, inner loop berjalan 1 kali iterasi.  
Pada iterasi kelima, inner loop tidak berjalan karena elemen terakhir sudah di tempatnya.
  - Selection Sort
    - Outer loop : 5 iterasi
    - Inner loop :  
Pada iterasi pertama, inner loop berjalan 4 kali.  
Pada iterasi kedua, inner loop berjalan 3 kali.  
Pada iterasi ketiga, inner loop berjalan 2 kali.  
Pada iterasi keempat, inner loop berjalan 1 kali.  
Pada iterasi kelima, inner loop tidak berjalan karena array sudah diurutkan.
  - Insertion Sort
    - Outer loop : 5 iterasi
    - Inner loop :  
Pada iterasi pertama, inner loop berjalan 1 kali.  
Pada iterasi kedua, inner loop berjalan 2 kali.  
Pada iterasi ketiga, inner loop berjalan 3 kali.  
Pada iterasi keempat, inner loop berjalan 4 kali.  
Pada iterasi kelima, inner loop berjalan 5 kali.
2. Untuk pengurutan secara ascending, elemen manakah ditemukan pada iterasi pertama untuk setiap jenis sorting? Apakah elemen terbesar atau elemen terkecil?  
**Jawab :**
    - Bubble Sort : Elemen ditemukan pada iterasi pertama adalah elemen terbesar. Bubble Sort bekerja dengan membandingkan elemen bersebelahan dan menukarnya jika mereka dalam urutan yang salah. Dalam iterasi pertama, elemen terbesar akan berada di posisi akhir array.
    - Selection Sort : Elemen ditemukan pada iterasi pertama adalah elemen terkecil.
    - Insertion Sort : Elemen ditemukan pada iterasi pertama adalah elemen yang berada pada elemen pertama pada array.
  3. Pada bubbleSort() terdapat kode sebagai berikut

```

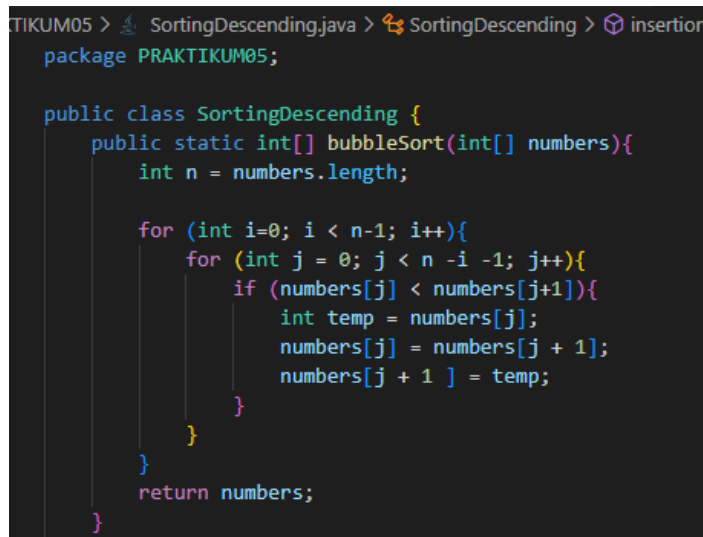
if (numbers[j - 1] > numbers[j]) {
    temp = numbers[j - 1];
    numbers[j - 1] = numbers[j];
    numbers[j] = temp;
}

```

Apa yang dilakukan oleh kode program tersebut?

**Jawab :** Membandingkan elemen bersebelahan dan menukarnya jika mereka dalam urutan yang salah. Kode `if (numbers[j] > numbers[j+1])` memeriksa apakah elemen pada indeks `j` lebih besar dari elemen berikutnya pada indeks `j+1`. Ini adalah langkah pertama dalam proses perbandingan dan penukaran elemen. Jika kondisi di atas terpenuhi (yaitu, elemen pada indeks `j` lebih besar dari elemen pada indeks `j+1`), maka kode tersebut menukar elemen tersebut. Ini dilakukan dengan menggunakan variabel sementara `temp` untuk menyimpan nilai elemen pada indeks `j`, kemudian mengatur elemen pada indeks `j` dengan nilai elemen pada indeks `j+1`, dan akhirnya mengatur elemen pada indeks `j+1` dengan nilai yang disimpan dalam `temp`.

4. Buatlah file baru `SortingDescending.java` kemudian buatlah 3 method untuk melakukan sorting secara descending



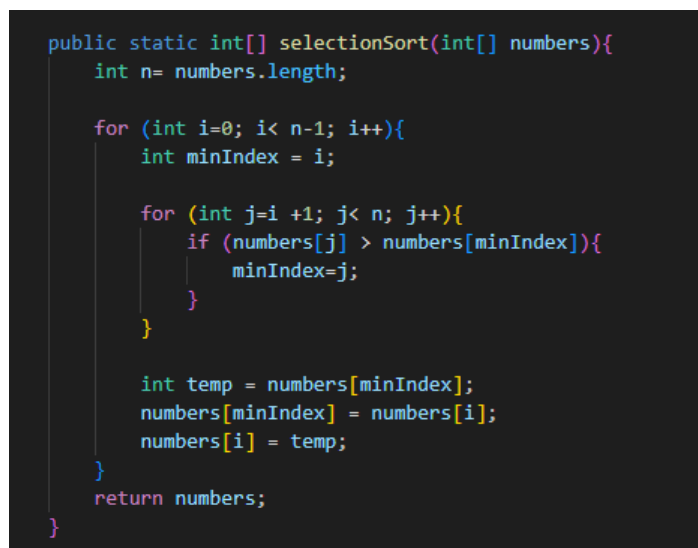
```

TIKUM05 > SortingDescending.java > SortingDescending > insertion
package PRAKTIKUM05;

public class SortingDescending {
    public static int[] bubbleSort(int[] numbers){
        int n = numbers.length;

        for (int i=0; i < n-1; i++){
            for (int j = 0; j < n -i -1; j++){
                if (numbers[j] < numbers[j+1]){
                    int temp = numbers[j];
                    numbers[j] = numbers[j + 1];
                    numbers[j + 1] = temp;
                }
            }
        }
        return numbers;
    }
}

```



```

public static int[] selectionSort(int[] numbers){
    int n= numbers.length;

    for (int i=0; i< n-1; i++){
        int minIndex = i;

        for (int j=i +1; j< n; j++){
            if (numbers[j] > numbers[minIndex]){
                minIndex=j;
            }
        }

        int temp = numbers[minIndex];
        numbers[minIndex] = numbers[i];
        numbers[i] = temp;
    }
    return numbers;
}

```

```

public static int[] insertionSort(int[] numbers){
    int n = numbers.length;

    for (int i=1; i < n; i++){
        int key = numbers[i];
        int j = i -1;

        while (j >= 0 && numbers[j] < key){
            numbers[j+1] = numbers[j];
            j = j-1;
        }

        numbers[j+1] = key;
    }
    return numbers;
}

```

```

KUM05 > SortingMain.java > SortingMain > main(String[])
public class SortingMain {
    public static void main(String[] args) {
        int my1stNumbers[] = { 8,6,10,2,4 };
        my1stNumbers = SortingAngka.bubbleSort(my1stNumbers);
        SortingAngka.printArray(my1stNumbers);

        int my2ndNumbers[] = { 7,9,3,11,1 };
        my2ndNumbers = SortingAngka.selectionSort(my2ndNumbers);
        SortingAngka.printArray(my2ndNumbers);

        int my3rdNumbers[] = { 4,1,12,6,5 };
        my3rdNumbers = SortingAngka.selectionSort(my3rdNumbers);
        SortingAngka.printArray(my3rdNumbers);

        System.out.println(x:"Sorting secara Descendig");
        my1stNumbers = SortingDescending.bubbleSort(my1stNumbers);
        SortingDescending.printArray(my1stNumbers);
        my2ndNumbers = SortingDescending.selectionSort(my2ndNumbers);
        SortingDescending.printArray(my2ndNumbers);
        my3rdNumbers = SortingDescending.selectionSort(my3rdNumbers);
        SortingDescending.printArray(my3rdNumbers);
    }
}

```

Sorting secara Descendig

10 8 6 4 2

11 9 7 3 1

12 6 5 4 1

PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>

main\* 0 8 0 Java: Ready

## Percobaan 2

### Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort

#### 5.4 Latihan 1

Tambahkan method pada class Kelas sebagai berikut:

1. selectionSortByUmur() untuk melakukan pengurutan berdasarkan umur menggunakan metode selection sort

```

public void selectionSortByUmur(){
    int n= daftarMahasiswa.length;

    for (int i=0; i< n-1; i++){
        int minIndex = i;

        for (int j=i +1; j< n; j++){
            if (daftarMahasiswa[j].umur < daftarMahasiswa[minIndex].umur){
                minIndex=j;
            }
        }

        Mahasiswa temp = daftarMahasiswa[minIndex];
        daftarMahasiswa[minIndex] = daftarMahasiswa[i];
        daftarMahasiswa[i] = temp;
    }
}

```

2. insertionSortByIPKDesc() untuk melakukan pengurutan berdasarkan IPK secara descending menggunakan metode insertion sort

```

public void insertionSortByIPKDesc(){
    int n = daftarMahasiswa.length;

    for (int i=1; i < n; i++){
        Mahasiswa key = daftarMahasiswa[i];
        int j = i -1;

        while (j >= 0 && daftarMahasiswa[j].ipk < key.ipk){
            daftarMahasiswa[j+1] = daftarMahasiswa[j];
            j = j-1;
        }

        daftarMahasiswa[j+1] = key;
    }
}

```

### 5.5 Latihan 2

Sebuah platform travel yang menyediakan layanan reservasi hotel. Hotel yang tersedia memiliki informasi nama, harga, dan rating (1-5). Tambahkan 5 data hotel. Daftar hotel ini harus dapat di sorting berdasarkan

1. Harga dimulai dari harga termurah ke harga tertinggi.
2. Rating bintang penginapan dari bintang tertinggi (5) ke terendah (1)

Buatlah proses sorting data untuk kedua filter tersebut dengan menggunakan algoritma bubble sort (No 1) dan selection sort (2).

```

PRAKTIKUM05 > Hotel.java > Hotel
package PRAKTIKUM05;

public class Hotel {
    public String nama;
    public int harga;
    public int rating;

    public Hotel(String nama, int harga, int rating) {
        this.nama = nama;
        this.harga = harga;
        this.rating = rating;
    }

    public void displayInfo() {
        System.out.println("Hotel (" +
            "nama='" + nama + '\'' +
            ", price=" + harga +
            ", rating=" + rating +
            ')');
    }
}

```

```

PRAKTIKUM05 > Hotels.java > Hotels > displayInfo()
package PRAKTIKUM05;

public class Hotels {
    Hotel [] dataHotels;
    int index;

    public Hotels(int hotel ){
        dataHotels = new Hotel[hotel];
    }

    public void add(Hotel daftarHotels){
        if (index < dataHotels.length){
            dataHotels[index] = daftarHotels;
            index++;
        }else {
            System.out.println(x:"Data Hotel tidak tersedia");
        }
    }

    public void displayInfo(){
        for (Hotel daftarHotel : dataHotels){
            daftarHotel.displayInfo();
        }
    }
}

```

IKUM05 > Hotels.java > Hotels > displayInfo()

```
public class Hotels {  
    }  
  
    public void bubbleSortHotelsByHarga() {  
        int n = dataHotels.length;  
        for (int i = 0; i < n - 1; i++) {  
            for (int j = 0; j < n - i - 1; j++) {  
                if (dataHotels[j].harga > dataHotels[j + 1].harga) {  
                    Hotel temp = dataHotels[j];  
                    dataHotels[j] = dataHotels[j + 1];  
                    dataHotels[j + 1] = temp;  
                }  
            }  
        }  
    }  
  
    public void selectionSortHotelsByRating() {  
        int n = dataHotels.length;  
        for (int i = 0; i < n - 1; i++) {  
            int minIndex = i;  
            for (int j = i + 1; j < n; j++) {  
                if (dataHotels[j].rating > dataHotels[minIndex].rating) {  
                    minIndex = j;  
                }  
            }  
            if (minIndex != i) {  
                Hotel temp = dataHotels[i];  
                dataHotels[i] = dataHotels[minIndex];  
                dataHotels[minIndex] = temp;  
            }  
        }  
    }  
}
```

```
public class HotelMain {  
    Run | Debug  
    public static void main(String[] args) {  
        Hotels dataHotel = new Hotels(hotel:5);  
        Hotel hotel1 = new Hotel(nama:"Hotel A", harga:100, rating:2);  
        Hotel hotel2 = new Hotel(nama:"Hotel B", harga:200, rating:5);  
        Hotel hotel3 = new Hotel(nama:"Hotel C", harga:150, rating:3);  
        Hotel hotel4 = new Hotel(nama:"Hotel D", harga:250, rating:1);  
        Hotel hotel5 = new Hotel(nama:"Hotel E", harga:300, rating:4);  
  
        dataHotel.add(hotel1);  
        dataHotel.add(hotel2);  
        dataHotel.add(hotel3);  
        dataHotel.add(hotel4);  
        dataHotel.add(hotel5);  
  
        dataHotel.bubbleSortHotelsByHarga();  
        dataHotel.selectionSortHotelsByRating();  
        dataHotel.displayInfo();  
    }  
}
```

pData\Roaming\Code\User\workspaceStorage\eb086

Hotel (nama='Hotel B', price=200, rating=5)

Hotel (nama='Hotel E', price=300, rating=4)

Hotel (nama='Hotel C', price=150, rating=3)

Hotel (nama='Hotel A', price=100, rating=2)

Hotel (nama='Hotel D', price=250, rating=1)

PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD> █

main\* ↺ ⊗ 0 ▲ 8 🔊 0 ➤ ☕ Java: Ready