

Nama : Rachmad Aprisandhy
Kelas : TI-1B
no Absen : 18

JOBSHEET 6 ASD SORTING

1. Praktikum 1

- a) Buat Clas Bernama Sorting no absen

J Sorting18.java 2 X

```
public class Sorting18{  
    int [] data ;  
    int jumData;
```

- b) Buat konstruktor dengan parameter data[] JumData

```
Sorting18 (int Data[], int JmlhData){  
    jumData = JmlhData;  
    data= new int[JmlhData];  
    for (int i=0; i<jumData;i++){  
        data[i]= Data [i];  
    }  
}
```

- c) Buat metode bubblesort

```
void bubblesort(){  
    int temp=0;  
    for (int i=0; i<jumData-1;i++){  
        for(int j=1; j< jumData-i;j++){  
            if (data[j-1]> data[j]){  
                temp=data[j];  
                data[j]=data[j-1];  
                data[j-1]=temp;  
            }  
        }  
    }  
}
```

- d) Buat method tampil

```
void tampil(){
    for(int i=0; i<jumData;i++){
        System.out.print(data[i] + "");
    }
    System.out.println();
}
```

- e) Buat class lagi dengan nama SortingMain no absen

```
import java.util.Scanner;
public class SortingMain18{
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
```

- f) Deklarasikan array dengan nama a

```
int a[] = {20,10,2,7,12};
```

- g) Buat objek baru bernama dataurut1 sebagai instansiasi dari class Sorting no Absen

```
Sorting18 dataurut1 = new Sorting18(a,a.length);
```

- h) Lakukan pemanggilan method

```
System.out.println(x:"data awal 1");
dataurut1.tampil();
dataurut1.bubblesort();
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
dataurut1.tampil();
```

- i) Cocokkan hasilnya

```
data awal 1
20102712
Data sudah diurutkan dengan BUBBLE SORT (ASC)
27101220
PS C:\Users\USER\OneDrive\Dokumen\ASD\Jobsheet 6> █
```

2. Selection Sort

- a) Tambahkan method baru yaitu selection sort

```
void selectionsort(){  
    for (int i=0; i< jumData; i++){  
        int min=i;  
        for (int j=i+1; j<jumData; j++){  
            if (data[j] < data [min]){  
                min=j;  
            }  
        }  
        int temp=data[i];  
        data[i]=data[min];  
        data[min]=temp;  
    }  
}
```

- b) Deklarasikan array dengan nama b

```
Sorting18 dataurut2 = new Sorting18(b,b.length);
```

- c) Lakukan pemanggilan Kembali method

```
System.out.println(x:"data awal 2");  
dataurut2.tampil();  
dataurut2.bubblesort();  
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");  
dataurut2.tampil();
```

- d) Cocokan hasilnya

```
data awal 2  
30202814  
Data sudah diurutkan dengan BUBBLE SORT (ASC)  
28142030  
PS C:\Users\USER\OneDrive\Dokumen\ASD\Jobsheet 6> 
```

3. Insertion sort

- a) Tambahkan method baru yaitu insertion sort

```
void insertionSort(){
    for(int i=1 ; i<=data.length-1;i++){
        int temp= data [i];
        int j=i-1;
        while (j>=0 && data [j] > temp){
            data [j+1]= data [j];
            j--;
        }
        data[j+1]=temp;
    }
}
```

- b) Deklarasikan array dengan nama C

```
int c[] = {40 , 10 , 4 , 9 , 3};
```

- c) Lakukan pemanggilan Kembali method

```
System.out.println(x:"data awal 3");
dataurut3.tampil();
dataurut3.insertionSort();
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
dataurut3.tampil();
```

- d) Cocokan hasilnya

```
data awal 3
4010493
Data sudah diurutkan dengan BUBBLE SORT (ASC)
3491040
```

1. Pertanyaan

- i. Jelaskan kode program berikut

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

- ii. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection short
- iii. Pada insertion short jelaskan maksud dairi kondisi pada perulangann while (j >= 0 && data[j] > temp)
- iv. Pada insertion sort apakah tujuan dari perintah data[j + 1] = data [j] ;

2. Jawaban

- i. Fungsi kode itu adalah untuk menukar elemen dalam sebuah array yang nantinya disusun dari kecil terlebih dahulu atau dari besar terlebih dahulu biasanya ini digunakan pada bubblesort
- ii. $J=i-1$
- iii. digunakan dalam Insertion Sort untuk memastikan elemen yang lebih besar digeser ke kanan.
- iv. digunakan untuk **menggeser elemen ke kanan** agar elemen baru bisa masuk pada posisi yang benar.

4. Praktikum 2

- a) Buat class baru yang bernama Mahasiswa no absen dan berikan juga konstruktornya

```
public class Mahasiswa18{
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa18(){

    }

    Mahasiswa18(String nm , String name , String kls ,double ip){
        nim =nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }

    void tampilInformasi(){
        System.out.println("nama : " + nama);
        System.out.println("NIM : " + nim);
        System.out.println("Kelas : " + kelas);
        System.out.println("IPK : " + ipk);
    }
}
```

- b) Buat class baru bernama mahasiswaBerprestasi no absen

```
public class MahasiswaBerprestasi18{
    Mahasiswa18 [] listMHS = new Mahasiswa18[5];
    int idx;
```

- c) Tambahkan method tambah

```
void tambah (Mahasiswa18 m){
    if (idx<listMHS.length){
        listMHS[idx]=m;
        idx++;
    }else{
        System.out.println(x:"data sudah penuh");
    }
}
```

- d) Tambahkan method tampil

```
void tampil(){
    for (Mahasiswa18 m : listMHS){
        m.tampilInformasi();
        System.out.println(x:"-----");
    }
}
```

- e) Tambahkan method bubblesort

```
void bubblesort(){
    for (int i = 0; i < listMHS.length - 1; i++){
        for (int j = 1; j < listMHS.length - i; j++) {
            if (listMHS[j].ipk > listMHS[j - 1].ipk) {
                Mahasiswa18 tmp = listMHS[j];
                listMHS[j] = listMHS[j - 1];
                listMHS[j - 1] = tmp;
            }
        }
    }
}
```

- f) Buat class baru Bernama Mahasiswa demo

```
public class MahasiswaDemo18{
```

- g) Buat objek mahasiswa berjumlah 5 buag tambahkan fungsi tambah pada mahasiswaberprestasi18

```
public static void main(String[] args) {
    MahasiswaBerprestasi18 list = new MahasiswaBerprestasi18();
    Mahasiswa18 m1 = new Mahasiswa18( nm:"123", name:"Zidan", kls:"2A", ip:3.2);
    Mahasiswa18 m2 = new Mahasiswa18( nm:"124", name:"Ayu", kls:"2A", ip:3.5);
    Mahasiswa18 m3 = new Mahasiswa18( nm:"125", name:"Sofi", kls:"2A", ip:3.1);
    Mahasiswa18 m4 = new Mahasiswa18( nm:"126", name:"Sita", kls:"2A", ip:3.9);
    Mahasiswa18 m5 = new Mahasiswa18( nm:"127", name:"miki", kls:"2A", ip:3.7);

    list.tambah(m1);
    list.tambah(m2);
    list.tambah(m3);
    list.tambah(m4);
    list.tambah(m5);
}
```

h) Panggil fungsi tampil

```
System.out.println(x:"Data mahasiswa Sebelum sorting : ");  
list.tampil();
```

i) Panggil fungsi bubblesort lalu tampilkan lagi

```
System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK (DESC) : ");  
list.bubblesort();  
list.tampil();|
```

j) Cocokan fungsinya

```
Data mahasiswa Sebelum sorting :
```

```
nama : Zidan
```

```
NIM : 123
```

```
Kelas : 2A
```

```
IPK : 3.2
```

```
-----  
nama : Ayu
```

```
NIM : 124
```

```
Kelas : 2A
```

```
IPK : 3.5
```

```
-----  
nama : Sofi
```

```
NIM : 125
```

```
Kelas : 2A
```

```
IPK : 3.1
```

```
-----  
nama : Sita
```

```
NIM : 126
```

```
Kelas : 2A
```

```
IPK : 3.9
```

```
-----  
nama : miki
```

```
NIM : 127
```

```
Kelas : 2A
```

```
IPK : 3.7
```



```
-----  
Data Mahasiswa setelah sorting berdasarkan IPK (DESC) :
```

```
nama : Sita
```

```
NIM : 126
```

```
Kelas : 2A
```

```
IPK : 3.9
```

```
-----  
nama : miki
```

```
NIM : 127
```

```
Kelas : 2A
```

```
IPK : 3.7
```

```
-----  
nama : Ayu
```

```
NIM : 124
```

```
Kelas : 2A
```

```
IPK : 3.5
```

```
-----  
nama : Zidan
```

```
NIM : 123
```

```
Kelas : 2A
```

```
IPK : 3.2
```

```
-----  
nama : Sofi
```

```
NIM : 125
```

```
Kelas : 2A
```

```
IPK : 3.1
```

```
-----  
PS C:\Users\USER\OneDrive\Dokumen\ASD\Jobsheet 6> █
```

5. Jawaban pertanyaan 2

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:

```
for (int i=0; i<listMhs.length-1; i++){  
    for (int j=1; j<listMhs.length-i; j++){
```

a. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?

- Perulangan i mengontrol jumlah iterasi utama dalam *Bubble Sort* dan Proses *Bubble Sort* membutuhkan maksimal $n-1$ iterasi untuk memastikan semua elemen sudah tersusun dengan benar. Oleh karena

itu, batas atas i adalah `listMhs.length - 1` agar tidak melakukan iterasi yang tidak perlu.

- b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - 1$?
- Perulangan j digunakan untuk membandingkan elemen dan melakukan pertukaran jika diperlukan dan setiap iterasi i membuat elemen terbesar pada bagian yang belum terurut berpindah ke posisi yang benar. Oleh karena itu, perulangan j tidak perlu menelusuri elemen yang sudah terurut, sehingga batasnya berkurang sebanyak i .
- c. Jika banyak data di dalam `listMhs` adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?
- Perulangan i akan berlangsung sebanyak $\text{listMhs.length} - 1 = 50 - 1 = 49$ kali dan bubble Sort bekerja dalam $n-1$ tahap, jadi dalam hal ini terdapat 49 tahap untuk mengurutkan seluruh elemen.

2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyborad) yang terdiri dari nim, nama, kelas, dan ipk!

```
import java.util.Scanner;

public class MahasiswaDemo18 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        MahasiswaBerprestasi18 list = new MahasiswaBerprestasi18();

        System.out.print(s:"Masukkan Jumlah Mahasiswa: ");
        int jmlmhs = input.nextInt();
        input.nextLine();

        for (int i = 0; i < jmlmhs; i++) {
            System.out.println("Masukkan Data mahasiswa ke - " + (i + 1));

            System.out.print(s:"NIM : ");
            String nim = input.nextLine();

            System.out.print(s:"Nama : ");
            String nama = input.nextLine();

            System.out.print(s:"Kelas : ");
            String kelas = input.nextLine();

            System.out.print(s:"IPK : ");
            double ipk = input.nextDouble();
            input.nextLine();

            Mahasiswa18 MHA = new Mahasiswa18(nim, nama, kelas, ipk);
            list.tambah(MHA);
        }

        System.out.println(x:"Data mahasiswa sebelum sorting:");
        list.tampil();
    }
}
```

6. Selection sort

- 1) Tambahkan method selection sort pada Mahasiswa berprestasi

```
void selectionsort(){
    for (int i=0; i<listMHS.length-1; i++){
        int idxmin=i;
        for (int j=i+1; j<listMHS.length; j++) {
            if (listMHS[j].ipk<listMHS[idxmin].ipk){
                idxmin=j;
            }
        }
        Mahasiswa18 tmp = listMHS[idxmin];
        listMHS[idxmin]=listMHS[i];
        listMHS[i] = tmp;
    }
}
```

- 2) Tambahkan baris program untuk memanggil method selection sort

```
System.out.println(x:"Data mahasiswa setelah sorting melalui selectionsort(ascending) : ");
list.selectionsort();
list.tampil();
```

3) Cocokan hasilnya

Data mahasiswa Sebelum sorting :

nama : Zidan

NIM : 123

Kelas : 2A

IPK : 3.2

nama : Ayu

NIM : 124

Kelas : 2A

IPK : 3.5

nama : Sofi

NIM : 125

Kelas : 2A

IPK : 3.1

nama : Sita

NIM : 126

Kelas : 2A

IPK : 3.9

nama : miki

NIM : 127

Kelas : 2A

IPK : 3.7

Data mahasiswa setelah sorting melalui selectionsort(ascending) :

nama : Sofi

NIM : 125

Kelas : 2A

IPK : 3.1

nama : Zidan

NIM : 123

Kelas : 2A

IPK : 3.2

nama : Ayu

NIM : 124

Kelas : 2A

IPK : 3.5

nama : miki

NIM : 127

Kelas : 2A

IPK : 3.7

nama : Sita

NIM : 126

Kelas : 2A

IPK : 3.9

7. Jawaban

```
int idxMin=i;
for (int j=i+1; j<listMhs.length; j++){
    if (listMhs[j].ipk<listMhs[idxMin].ipk){
        idxMin=j;
    }
}
```

menyimpan indeks sementara dari elemen dengan IPK terkecil dalam iterasi ini. Awalnya, indeks i dianggap sebagai nilai minimum.yang nantinya jika ditemukan elemen dengan indeks lebih kecil maka akan diperbarui menjadi elemen tersebut

8. Insertion sort

- 1) Buat method baru insertion sort pada mahasiswa berprestasi

```
void insertionsort(){
    for(int i=1 ; i<listMHS.length;i++){
        Mahasiswa18 temp= listMHS[i];
        int j=i;
        while (j>0 && listMHS [j-1].ipk>temp.ipk)
            listMHS[j] = listMHS[j-1];
            j--;
        }
        listMHS[j] = temp;
    }
}
```

- 2) Panggil method insertion sort pada mahasiswa demo

```
System.out.println(x:"data yang sudah terurut menggunakan Inssertion sort (ASC)");
list.insertionsort();
list.tampil();|
```

3) Cocokan hasilnya

Data mahasiswa Sebelum sorting :

nama : Zidan

NIM : 123

Kelas : 2A

IPK : 3.2

nama : Ayu

NIM : 124

Kelas : 2A

IPK : 3.5

nama : Sofi

NIM : 125

Kelas : 2A

IPK : 3.1

nama : Sita

NIM : 126

Kelas : 2A

IPK : 3.9

nama : miki

NIM : 127

Kelas : 2A

IPK : 3.7

```
data yang sudah terurut menggunakan Insertion sort (ASC)
nama : Sofi
NIM : 125
Kelas : 2A
IPK : 3.1
-----
nama : Zidan
NIM : 123
Kelas : 2A
IPK : 3.2
-----
nama : Ayu
NIM : 124
Kelas : 2A
IPK : 3.5
-----
nama : miki
NIM : 127
Kelas : 2A
IPK : 3.7
-----
nama : Sita
NIM : 126
Kelas : 2A
IPK : 3.9
```

9. Jawaban pertanyaan

Ubahlah fungsi pada insertionsort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending

- Hanya dengan mengubah **while (j>0 && listMHS [j-1].ipk>temp.ipk){** menjadi **while (j>0 && listMHS [j-1].ipk<temp.ipk){**

10. Tugas

```
public class DataDosen {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    public DataDosen() {
    }

    public DataDosen(String kode, String nama, boolean jenisKelamin, int usia) {
        this.kode = kode;
        this.nama = nama;
        this.jenisKelamin = jenisKelamin;
        this.usia = usia;
    }

    public void tampilInformasi() {
        System.out.println("Kode : " + kode);
        System.out.println("Nama : " + nama);
        System.out.println("Jenis Kelamin : " + (jenisKelamin ? "Laki-laki" : "Perempuan"));
        System.out.println("Usia : " + usia);
        System.out.println(x: "-----");
    }
}
```

```

1 public class DosenDemo {
2     DataDosen[] listDSN;
3     int jumlahData;
4
5     public DosenDemo(int kapasitas) {
6         listDSN = new DataDosen[kapasitas];
7         jumlahData = 0;
8     }
9
10    public void tambah(DataDosen d) {
11        if (jumlahData < listDSN.length) {
12            listDSN[jumlahData] = d;
13            jumlahData++;
14        } else {
15            System.out.println(x:"Data sudah penuh!");
16        }
17    }
18
19    public void tampil() {
20        if (jumlahData == 0) {
21            System.out.println(x:"Belum ada data dosen.");
22        } else {
23            for (int i = 0; i < jumlahData; i++) {
24                listDSN[i].tampilInformasi();
25            }
26        }
27    }
28
29    public void bubblesort() {
30        for (int i = 0; i < jumlahData - 1; i++) {
31            for (int j = 0; j < jumlahData - 1 - i; j++) {
32                if (listDSN[j].usia > listDSN[j + 1].usia) {
33                    DataDosen temp = listDSN[j];
34                    listDSN[j] = listDSN[j + 1];
35                    listDSN[j + 1] = temp;
36                }
37            }
38        }
39    }
40 }

```

```
public void selectionsort() {
    for (int i = 0; i < jumlahData - 1; i++) {
        int maxIndex = i;
        for (int j = i + 1; j < jumlahData; j++) {
            if (listDSN[j].usia > listDSN[maxIndex].usia) {
                maxIndex = j;
            }
        }
        DataDosen temp = listDSN[maxIndex];
        listDSN[maxIndex] = listDSN[i];
        listDSN[i] = temp;
    }
}

public void insertionsort() {
    for (int i = 1; i < jumlahData; i++) {
        DataDosen temp = listDSN[i];
        int j = i;
        while (j > 0 && listDSN[j - 1].usia < temp.usia) {
            listDSN[j] = listDSN[j - 1];
            j--;
        }
        listDSN[j] = temp;
    }
}
}
```

```

import java.util.Scanner;

public class DosenMain18 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        DosenDemo list = new DosenDemo(kapasitas:10);

        while (true) {
            System.out.println(x:"==== Menu Pilihan ===");
            System.out.println(x:"1. Tambah Data Dosen");
            System.out.println(x:"2. Tampilkan Data Dosen");
            System.out.println(x:"3. Mengurutkan data dosen (ASC) Bubble Sort");
            System.out.println(x:"4. Mengurutkan data dosen (DESC) Selection Sort");
            System.out.println(x:"5. Mengurutkan data dosen (ASC) Insertion Sort");
            System.out.println(x:"6. Keluar");
            System.out.println(x:"=====");

            System.out.print(s:"Masukkan pilihan anda: ");
            int pilihan = input.nextInt();
            input.nextLine();

            if (pilihan == 1) {
                System.out.print(s:"Masukkan Kode: ");
                String kode = input.nextLine();
                System.out.print(s:"Masukkan Nama: ");
                String nama = input.nextLine();
                System.out.print(s:"Masukkan jenis kelamin (m/f): ");
                char jk = input.next().charAt(index:0);
                boolean jenisKelamin = (jk == 'm' || jk == 'M');
                System.out.print(s:"Masukkan usia: ");
                int usia = input.nextInt();
                input.nextLine();
            }
        }
    }
}

```

```
        DataDosen DSN = new DataDosen(kode, nama, jenisKelamin, usia);
        list.tambah(DSN);
        System.out.println(x:"Data Dosen Ditambahkan!");

    } else if (pilihan == 2) {
        System.out.println(x:"Daftar Dosen:");
        list.tampil();
    } else if (pilihan == 3) {
        list.bubblesort();
        System.out.println(x:"Data diurutkan dari usia terkecil.");
        list.tampil();
    } else if (pilihan == 4) {
        list.selectionsort();
        System.out.println(x:"Data diurutkan dari usia terbesar.");
        list.tampil();
    } else if (pilihan == 5) {
        list.insertionsort();
        System.out.println(x:"Data diurutkan dari usia terbesar.");
        list.tampil();
    } else if (pilihan == 6) {
        System.out.println(x:"Program selesai.");
        break;
    } else {
        System.err.println(x:"Pilihan tidak valid!");
    }
}
input.close();
}
```