

Laporan Hasil Praktikum ALSD 6



NAMA : Arifah Zhafirah Wikananda

NIM : 244107020188

KELAS : 1E

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2025

5.2.1 Langkah Praktikum 1

a. SORTING – BUBBLE SORT

```
public class Sorting04 {
    int [] data;
    int jumData;

    Sorting04 (int Data [], int jmlDat){
        jumData = jmlDat;
        data = new int[jmlDat];
        for (int i = 0; i < jumData; i++){
            data[i] = Data[i];
        }
    }

    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;
                }
            }
        }
    }

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

```
public class SortingMain04 {
    public static void main(String[] args) {
        int a [] = {20, 10, 2, 7, 12};
        Sorting04 dataurut1 = new Sorting04(a, a.length);

        System.out.println("Data awal 1");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBLE SORT (ASC)");
        dataurut1.tampil();
    }
}
```

Outputnya:

```
dk\java\latest\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\WINDOWS 11\AppData\Roaming\Cod
e\User\workspaceStorage\d83555ec4f9c765b284f331a909f075b\redhat.java\jdt_ws\Praktikum05_b1713034\bin" SortingMain04 "
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
```

b. SORTING – SELECTION SORT

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

```
int b[] = {30, 20, 2, 8, 14};
Sorting04 dataurut2 = new Sorting04(b, b.length);

System.out.println("Data awal 2");
dataurut2.tampil();
dataurut2.bubbleSort();
System.out.println("Data sudah diurutkan dengan SELECTION SORT (ASC)");
dataurut2.tampil();
```

Outputnya:

```
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
```

c. SORTING – INSERTION SORT

```
void insertionSort(){
    for (int i = 0; i < jumData-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;
    }
}
```

```
int c[] = {40, 10, 4, 9, 3};
    Sorting04 dataurut3 = new Sorting04(c, c.length);

    System.out.println("Data awal 3");
    dataurut3.tampil();
    dataurut3.bubbleSort();
    System.out.println("Data sudah diurutkan dengan SELECTION SORT (ASC)");
    dataurut3.tampil();
```

Outputnya:

```
Data awal 3
40 10 4 9 3
Data sudah diurutkan dengan SELECTION SORT (ASC)
3 4 9 10 40
```

Pertanyaan:

1. Jelaskan fungsi kode program berikut

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

Jawab:

1. Kondisi if ($\text{data}[j-1] > \text{data}[j]$)

- Mengecek apakah elemen sebelumnya ($\text{data}[j-1]$) lebih besar dari elemen saat ini ($\text{data}[j]$).
- Jika iya, maka kedua elemen ini harus ditukar agar berada dalam urutan yang benar (ascending).

2. Proses pertukaran (swap):

- Simpan $\text{data}[j]$ ke variabel sementara temp.
- Geser $\text{data}[j-1]$ ke posisi $\text{data}[j]$.
- Kembalikan temp ke posisi $\text{data}[j-1]$, sehingga elemen sebelumnya dan elemen saat ini bertukar posisi.

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection

Jawab:

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

3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan

```
while (j>=0 && data[j]>temp)
```

Jawab:

- **Kondisi ini memastikan elemen lebih besar digeser ke kanan** sebelum elemen baru (temp) dimasukkan ke posisi yang benar.
- Ini adalah **langkah utama dalam Insertion Sort** yang membuat array tetap terurut selama proses penyisipan.

4. Pada Insertion sort, apakah tujuan dari perintah

```
data[j+1] = data[j];
```

Jawab:

- **Perintah data[j+1] = data[j]; berfungsi untuk menggeser elemen ke kanan**, memberikan ruang bagi elemen yang lebih kecil untuk disisipkan.
- **Proses ini dilakukan berulang kali hingga elemen yang lebih kecil berada di posisi yang benar dalam array.**
- Ini adalah **langkah utama dalam Insertion Sort** yang memastikan array tetap terurut selama proses penyisipan elemen baru.

5.3.2 Langkah-langkah Praktikum 2

```
public class Mahasiswa04 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    // konstruktor default
    public Mahasiswa04() {
    }

    // Konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama
    ada yang tidak)
    public Mahasiswa04(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);
    }
}
```

```
public class MahasiswaBerprestasi04 {
    Mahasiswa04[] listMhs = new Mahasiswa04[5];
    int idx;

    void tambah (Mahasiswa04 m){
        if (idx < listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else{
            System.out.println("Data Sudah Penuh");
        }
    }

    void tampil(){
        for (Mahasiswa04 m:listMhs){
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    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){
                    Mahasiswa04 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}
```

```

public class MahasiswaDemo04 {
    public static void main(String[] args) {
        MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();
        Mahasiswa04 m1 = new Mahasiswa04("123", "Zidan", "2A", 3.2);
        Mahasiswa04 m2 = new Mahasiswa04("124", "Ayu", "2A", 3.5);
        Mahasiswa04 m3 = new Mahasiswa04("125", "Sofi", "2A", 3.1);
        Mahasiswa04 m4 = new Mahasiswa04("126", "Sita", "2A", 3.9);
        Mahasiswa04 m5 = new Mahasiswa04("127", "Miki", "2A", 3.7);

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

        System.out.println("Data Mahasiswa Sebelum Sorting: ");
        list.tampil();

        System.out.println("Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC): ");
        list.bubbleSort();
        list.tampil();
    }
}

```

Outputnya:

```

c:\Praktikum ASD\Praktikum05>
c:\Praktikum ASD\Praktikum05> c: && cd "c:\Praktikum ASD\Praktikum05" && cmd /C ""C:\Users\WINDOWS 11\AppData\Roaming\Code\Use
xtension-pack-jdk\java\latest\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\WINDOWS 11\
paceStorage\d83555ec4f9c765b284f331a909f075b\redhat.java\jdt_ws\Praktikum05_b1713034\bin" MahasiswaDemo04 "
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

Pertanyaan:

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 $i < \text{listMhs.length} - 1$?

Jawab:

- Perulangan i digunakan untuk melakukan iterasi sebanyak jumlah tahap dalam Bubble Sort.
- Pada setiap tahap, satu elemen terbesar akan "menggelembung" ke posisi yang benar.
- Setelah $n-1$ iterasi, seluruh elemen sudah terurut, karena elemen terakhir otomatis sudah berada di tempatnya.
- Jika $\text{listMhs.length} = n$, maka jumlah iterasi yang dibutuhkan adalah $n-1$ tahap, sehingga batas perulangan adalah $i < \text{listMhs.length} - 1$.

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

Jawab:

- Perulangan j digunakan untuk membandingkan dan menukar elemen jika diperlukan.
- j selalu lebih kecil dari $\text{listMhs.length} - i$ karena pada setiap tahap, elemen terbesar sudah berada di akhir array dan tidak perlu dibandingkan lagi.
- Dengan setiap iterasi i, jumlah elemen yang perlu dibandingkan berkurang sebanyak i karena elemen terbesar sudah berada di posisi yang benar.

c. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan

Jawab:

- berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?
- Jumlah iterasi i berlangsung sebanyak
 $\text{listMhs.length} - 1 = 50 - 1 = 49$
 $\text{listMhs.length} - 1 = 50 - 1 = 49$ Jadi, perulangan i akan berlangsung 49 kali.
- Tahapan Bubble Sort yang ditempuh sama dengan jumlah iterasi i, yaitu 49 tahap.

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

```
public class Mahasiswa04 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    // konstruktor default
    public Mahasiswa04() {
    }

    // Konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama ada yang tidak)
    public Mahasiswa04(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);
    }
}
```

```

public class MahasiswaBerprestasi04 {
    Mahasiswa04[] listMhs = new Mahasiswa04[5];
    int idx;

    void tambah (Mahasiswa04 m){
        if (idx < listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else{
            System.out.println("Data Sudah Penuh");
        }
    }

    void tampil(){
        for (Mahasiswa04 m:listMhs){
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    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){
                    Mahasiswa04 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}

```

```
import java.util.Scanner;

public class MahasiswaDemo04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();

        for (int i = 0; i < 5; i++) {
            System.out.println("Masukkan Mahasiswa ke-" + (i+1) + ":");
            System.out.print("NIM: ");
            String nim = sc.nextLine();
            System.out.print("Nama: ");
            String nama = sc.nextLine();
            System.out.print("Kelas: ");
            String kelas = sc.nextLine();
            System.out.print("IPK: ");
            double ipk = sc.nextDouble();
            System.out.println("-----");
            sc.nextLine();

            Mahasiswa04 m = new Mahasiswa04(nim, nama, kelas, ipk);
            list.tambah(m);

        }
        System.out.println("\nData mahasiswa sebelum sorting: ");
        list.tampil();

        System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC): ");
        list.bubbleSort();
        list.tampil();

        sc.close();
    }
}
```

Outputnya:

```
xtension-pack-jdk\java\latest\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -
paceStorage\d83555ec4f9c765b284f331a909f075b\redhat.java\jdt_ws\Praktikum05_b1713034\bin" MahasiswaDem
Masukkan Mahasiswa ke-1:
NIM: 123
Nama: rere
Kelas: 2e
IPK: 3,4
-----
Masukkan Mahasiswa ke-2:
NIM: 124
Nama: tete
Kelas: 2e
IPK: 3,5
-----
Masukkan Mahasiswa ke-3:
NIM: 126
Nama: yeye
Kelas: 2e
IPK: 3,2
-----
Masukkan Mahasiswa ke-4:
NIM: 127
Nama: wewe
Kelas: 2e
IPK: 3,1
-----
Masukkan Mahasiswa ke-5:
NIM: 128
Nama: fefe
Kelas: 2e
IPK: 3,7
-----
```

Data mahasiswa sebelum sorting:

Nama : rere
NIM : 123
Kelas : 2e
ipk : 3.4

Nama : tete
NIM : 124
Kelas : 2e
ipk : 3.5

Nama : yeye
NIM : 126
Kelas : 2e
ipk : 3.2

Nama : wewe
NIM : 127
Kelas : 2e
ipk : 3.1

Nama : fefe
NIM : 128
Kelas : 2e
ipk : 3.7

Data mahasiswa setelah sorting berdasarkan IPK (DESC):

Nama : fefe
NIM : 128
Kelas : 2e
ipk : 3.7

Data mahasiswa setelah sorting berdasarkan IPK (DESC):

Nama : fefe
NIM : 128
Kelas : 2e
ipk : 3.7

Nama : tete
NIM : 124
Kelas : 2e
ipk : 3.5

Nama : rere
NIM : 123
Kelas : 2e
ipk : 3.4

Nama : yeye
NIM : 126
Kelas : 2e
ipk : 3.2

Nama : wewe
NIM : 127
Kelas : 2e
ipk : 3.1

5.3.5 Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

```
public class MahasiswaBerprestasi04 {
    Mahasiswa04[] listMhs = new Mahasiswa04[5];
    int idx;

    void tambah (Mahasiswa04 m){
        if (idx < listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else{
            System.out.println("Data Sudah Penuh");
        }
    }

    void tampil(){
        for (Mahasiswa04 m:listMhs){
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    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){
                    Mahasiswa04 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa04 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
}
```

```

public class MahasiswaDemo04 {
    public static void main(String[] args) {
        MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();
        Mahasiswa04 m1 = new Mahasiswa04("123", "Zidan", "2A", 3.2);
        Mahasiswa04 m2 = new Mahasiswa04("124", "Ayu", "2A", 3.5);
        Mahasiswa04 m3 = new Mahasiswa04("125", "Sofi", "2A", 3.1);
        Mahasiswa04 m4 = new Mahasiswa04("126", "Sita", "2A", 3.9);
        Mahasiswa04 m5 = new Mahasiswa04("127", "Miki", "2A", 3.7);

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

        System.out.println("Data Mahasiswa Sebelum Sorting: ");
        list.tampil();

        System.out.println("Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC): ");
        list.bubbleSort();
        list.tampil();

        System.out.println("data yang sudah terurut menggunakan SELECTION SORT (ASC)");
        list.selectionSort();
        list.tampil();
    }
}

```

Outputnya:

```

c:\Praktikum ASD\Praktikum05>
c:\Praktikum ASD\Praktikum05> c: && cd "c:\Praktikum ASD\Praktikum05" && cmd /C ""C:\Users\WINDOWS 11\AppData\Roaming\Code\
xtension-pack-jdk\java\latest\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\WINDOWS
paceStorage\d83555ec4f9c765b284f331a909f075b\redhat.java\jdt_ws\Praktikum05_b1713034\bin" MahasiswaDemo04 "
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 SELECTION 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

Pertanyaan:

Di dalam method selection sort, terdapat baris program seperti di bawah ini:

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

Untuk apakah proses tersebut, jelaskan!

Jawab:

- ✓ Kode ini berfungsi untuk mencari indeks elemen dengan nilai terkecil dalam array.
- ✓ Digunakan dalam Selection Sort untuk menemukan elemen terkecil sebelum dilakukan pertukaran.
- ✓ Menjadi dasar dari proses pengurutan Selection Sort, yang bekerja dengan memilih elemen terkecil secara bertahap.

5.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

```
public class MahasiswaBerprestasi04 {
    Mahasiswa04[] listMhs = new Mahasiswa04[5];
    int idx;

    void tambah (Mahasiswa04 m){
        if (idx < listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else{
            System.out.println("Data Sudah Penuh");
        }
    }

    void tampil(){
        for (Mahasiswa04 m:listMhs){
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    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){
                    Mahasiswa04 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa04 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }

    void insertionSort(){
        for (int i = 1; i < listMhs.length; i++){
            Mahasiswa04 temp = listMhs[i];
            int j = i;
            while (j > 0 && listMhs[j-1].ipk > temp.ipk){
                listMhs[j] = listMhs[j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}
```

```

import java.util.Scanner;
public class MahasiswaDemo04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();
        Mahasiswa04 m1 = new Mahasiswa04("123", "Zidan", "2A", 3.2);
        Mahasiswa04 m2 = new Mahasiswa04("124", "Ayu", "2A", 3.5);
        Mahasiswa04 m3 = new Mahasiswa04("125", "Sofi", "2A", 3.1);
        Mahasiswa04 m4 = new Mahasiswa04("126", "Sita", "2A", 3.9);
        Mahasiswa04 m5 = new Mahasiswa04("127", "Miki", "2A", 3.7);

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

        System.out.println("Data Mahasiswa Sebelum Sorting: ");
        list.tampil();

        System.out.println("Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC): ");
        list.bubbleSort();
        list.tampil();

        System.out.println("data yang sudah terurut menggunakan SELECTION SORT (ASC)");
        list.selectionSort();
        list.tampil();

        System.out.println("Data yang sudah terurut menggunakan INSERTION SORT (ASC)");
        list.insertionSort();
        list.tampil();
    }
}

```

Outputnya:

```

xtension-pack-jdk\java\latest\bin\java.exe" --en
paceStorage\d83555ec4f9c765b284f331a909f075b\redh
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

Pertanyaan:

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

```
void insertionSort(){
    for (int i = 1; i < listMhs.length; i++){
        Mahasiswa04 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j-1].ipk < temp.ipk){
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Outputnya:

```
xtension-pack-jdk\java\latest\bin\java.exe" --enable-preview -XX:+Show
paceStorage\d83555ec4f9c765b284f331a909f075b\redhat.java\jdt_ws\Prakti
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 : 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
-----
```

5.5 Latihan Praktikum

Waktu : 45 Menit

Perhatikan class diagram dibawah ini:

Dosen
kode: String nama: String jenisKelamin: Boolean usia: int
Dosen(kd: String, name: String, jk: Boolean, age: int) tampil(): void

DataDosen
dataDosen: Dosen[10] idx: int
tambah(dsn: Dosen): void tampil(): void SortingASC(): void sortingDSC():void insertionSort():void

Berdasarkan class diagram diatas buatlah menu dikelas main dengan pilihan menu:

1. Tambah data digunakan untuk menambahkan data dosen
2. Tampil data digunakan untuk menampilkan data seluruh dosen
3. Sorting ASC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari dosen termuda ke dosen tertua menggunakan bubble Sort.
4. Sorting DSC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari tertua ke dosen termuda dapat menggunakan algoritma selection sort atau insertion sort.

Jawab:

```
public class dosen04 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    dosen04(String kd, String name, boolean jk, int age){
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

    void tampil(){
        System.out.println("Kode : " + kode);
        System.out.println("Nama : " + nama);
        System.out.println("Jenis Kelamin : " + jenisKelamin);
        System.out.println("Usia : " + usia);
    }
}
```

```
public class dataDosen04 {
    dosen04[] dataDosen = new dosen04[10];
    int idx;

    void tambah(dosen04 dsn){
        if (idx < dataDosen.length){
            dataDosen[idx] = dsn;
            idx++;
        }else {
            System.out.println("Data Dosen Sudah Banyak!");
        }
    }
    // Untuk menampilkan Data dosen
    void tampil (){
        if (idx == 0){
            System.out.println("Tidak Ada Data Dosen.");
            return;
        }
        for (int i = 0; i < idx; i++){
            dataDosen[i].tampil();
            System.out.println();
        }
    }
    // Metode untuk mengurutkan data dosen secara ascending (asc)
    void sortingASC(){
        for (int i = 0; i < idx - 1; i++){
            for (int j = 0; j < idx - 1 -i; j++){
                if (dataDosen[j].usia > dataDosen[j+1].usia){
                    dosen04 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
    }
}
```

```
// Metode untuk mengurutkan data dosen secara descending (dsc)
void sortingDSC(){
    for (int i = 0; i < idx - 1; i++){
        for (int j = 0; j < idx - 1 - i; j++){
            if (dataDosen[j].usia < dataDosen[j + 1].usia){
                dosen04 temp = dataDosen[j];
                dataDosen[j] = dataDosen[j + 1];
                dataDosen[j + 1] = temp;
            }
        }
    }
}
}
```

```
import java.util.Scanner;

public class dosenDemo04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        dataDosen04 datadsn = new dataDosen04();
        int pilihan;

        do {
            System.out.println("\nMenu");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Sorting ASC (Usia Termuda Ke Tertua)");
            System.out.println("4. Sorting DSC (Usia Tertua ke Termuda)");
            System.out.println("5. Keluar");
            System.out.print("Pilih Menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    for (int i = 0; i < 10; i++) {
                        System.out.println("Masukkan data dosen ke-" + (i + 1) + ":");
                        System.out.print("Masukkan kode dosen: ");
                        String kode = sc.nextLine();
                        System.out.print("Masukkan nama dosen: ");
                        String nama = sc.nextLine();
                        System.out.print("Masukkan jenis Kelamin (Perempuan / laki-laki): ");
                        char jk = sc.next().charAt(0);
                        Boolean jenisKelamin = (jk == 'l' || jk == 'p');
                        System.out.print("Masukkan usia dosen: ");
                        int usia = sc.nextInt();
                        sc.nextLine();
                        System.out.println("-----");

                        dosen04 dsn = new dosen04(kode, nama, jenisKelamin, usia);
                        datadsn.tambah(dsn);
                    }
                    break;
                case 2:
                    datadsn.tampil();
                    break;
                case 3:
                    datadsn.sortingASC();
                    System.out.println("Data Dosen Telah Diurutkan secara Ascending");
                    datadsn.tampil();
                    break;
                case 4:
                    break;
            }
        } while (pilihan != 5);
    }
}
```

```

        datadsn.sortingDSC();
        System.out.println("Data Dosen Telah Diurutkan secara Descending");
        datadsn.tampil();
        break;
    case 5:
        System.out.println("keluar dari Program");
        break;
    default:
        System.out.println("Pilihan tidak Valid. Silahkan Coba Lagi");
    }
}while (pilihan != 5);
}
}

```

Outputnya:

```

C:\Praktikum ASD\Praktikum05>
C:\Praktikum ASD\Praktikum05> c: && cd "c:\Praktikum ASD\Praktikum05" && java -jar extension-pack-jdk\java\latest\bin\java.exe --enable-preview -cp spaceStorage\d83555ec4f9c765b284f331a909f075b\redhat.java\
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda Ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Keluar
Pilih Menu: 1
Masukkan data dosen ke-1:
Masukkan kode dosen: 120
Masukkan nama dosen: pak hendra
Masukkan jenis Kelamin (Perempuan / laki-laki): 1
Masukkan usia dosen: 43
-----
Masukkan data dosen ke-2:
Masukkan kode dosen: 121
Masukkan nama dosen: pak budi
Masukkan jenis Kelamin (Perempuan / laki-laki): 1
Masukkan usia dosen: 45
-----
Masukkan data dosen ke-3:
Masukkan kode dosen: 122
Masukkan nama dosen: pak hapis
Masukkan jenis Kelamin (Perempuan / laki-laki): 1
Masukkan usia dosen: 46
-----

```

```

Masukkan data dosen ke-4:
Masukkan kode dosen: 124
Masukkan nama dosen: pak fauzi
Masukkan jenis Kelamin (Perempuan / laki-laki): 1
Masukkan usia dosen: 41
-----
Masukkan data dosen ke-5:
Masukkan kode dosen: 125
Masukkan nama dosen: pak angga
Masukkan jenis Kelamin (Perempuan / laki-laki): 1
Masukkan usia dosen: 47
-----
Masukkan data dosen ke-6:
Masukkan kode dosen: 123
Masukkan nama dosen: bu kayla
Masukkan jenis Kelamin (Perempuan / laki-laki): p
Masukkan usia dosen: 34
-----
Masukkan data dosen ke-7:
Masukkan kode dosen: 126
Masukkan nama dosen: bu angel
Masukkan jenis Kelamin (Perempuan / laki-laki): p
Masukkan usia dosen: 32
-----
Masukkan data dosen ke-8:
Masukkan kode dosen: 127
Masukkan nama dosen: bu vit
Masukkan jenis Kelamin (Perempuan / laki-laki): p
Masukkan usia dosen: 35
-----

```



```
Masukkan data dosen ke-9:
Masukkan kode dosen: 128
Masukkan nama dosen: bu vivi
Masukkan jenis Kelamin (Perempuan / laki-laki): p
Masukkan usia dosen: 31
-----
```

```
Masukkan data dosen ke-10:
Masukkan kode dosen: 129
Masukkan nama dosen: bu rani
Masukkan jenis Kelamin (Perempuan / laki-laki): p
Masukkan usia dosen: 37
-----
```

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda Ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Keluar

Pilih Menu: 2

```
Kode : 120
Nama : pak hendra
Jenis Kelamin : true
Usia : 43
```

```
Kode : 121
Nama : pak budi
Jenis Kelamin : true
Usia : 45
```

```
Kode : 128
Nama : bu vivi
Jenis Kelamin : true
Usia : 31
```

```
Kode : 129
Nama : bu rani
Jenis Kelamin : true
Usia : 37
```

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda Ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Keluar

Pilih Menu: 3

Data Dosen Telah Diurutkan secara Ascending

```
Kode : 128
Nama : bu vivi
Jenis Kelamin : true
Usia : 31
```

```
Kode : 126
Nama : bu angel
Jenis Kelamin : true
Usia : 32
```

```
Kode : 122
Nama : pak hapis
Jenis Kelamin : true
Usia : 46
```

```
Kode : 124
Nama : pak fauzi
Jenis Kelamin : true
Usia : 41
```

```
Kode : 125
Nama : pak angga
Jenis Kelamin : true
Usia : 47
```

```
Kode : 123
Nama : bu kayla
Jenis Kelamin : true
Usia : 34
```

```
Kode : 126
Nama : bu angel
Jenis Kelamin : true
Usia : 32
```

```
Kode : 127
Nama : bu vit
Jenis Kelamin : true
Usia : 35
```

```
Kode : 123
Nama : bu kayla
Jenis Kelamin : true
Usia : 34
```

```
Kode : 127
Nama : bu vit
Jenis Kelamin : true
Usia : 35
```

```
Kode : 129
Nama : bu rani
Jenis Kelamin : true
Usia : 37
```

```
Kode : 124
Nama : pak fauzi
Jenis Kelamin : true
Usia : 41
```

```
Kode : 120
Nama : pak hendra
Jenis Kelamin : true
Usia : 43
```

```
Kode : 121
Nama : pak budi
Jenis Kelamin : true
Usia : 45
```

```
Kode : 122
Nama : pak hapis
Jenis Kelamin : true
Usia : 46
```

```
Kode : 125
Nama : pak angga
Jenis Kelamin : true
Usia : 47
```

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda Ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Keluar

Pilih Menu: 4

Data Dosen Telah Diurutkan secara Descending

```
Kode : 125
Nama : pak angga
Jenis Kelamin : true
Usia : 47
```

```
Kode : 122
Nama : pak hapis
Jenis Kelamin : true
Usia : 46
```

Kode : 121
Nama : pak budi
Jenis Kelamin : true
Usia : 45

Kode : 120
Nama : pak hendra
Jenis Kelamin : true
Usia : 43

Kode : 124
Nama : pak fauzi
Jenis Kelamin : true
Usia : 41

Kode : 129
Nama : bu rani
Jenis Kelamin : true
Usia : 37

Kode : 127
Nama : bu vit
Jenis Kelamin : true
Usia : 35

Kode : 123
Nama : bu kayla
Jenis Kelamin : true
Usia : 34

Kode : 127
Nama : bu vit
Jenis Kelamin : true
Usia : 35

Kode : 123
Nama : bu kayla
Jenis Kelamin : true
Usia : 34

Kode : 126
Nama : bu angel
Jenis Kelamin : true
Usia : 32

Kode : 128
Nama : bu vivi
Jenis Kelamin : true
Usia : 31

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda Ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Keluar

Pilih Menu: 5

keluar dari Program