

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN STRUKTUR DATA  
JOBSHEET 6**



**RANI MIFTAHUS SAÁDAH**

**244107020057**

**TI\_1E**

**PROGRAM STUDI D\_IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
PRAKTIKUM 2025**

## 6.1.1 Mengimplementasikan Sorting menggunakan object

### a. SORTING – BUBBLE SORT

```
public class Sorting25 {

    int [] data;
    int jumData;

    Sorting25 (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 MainSorting25 {
    public static void main(String[] args) {
        int a[]={20, 10, 2, 7, 12};
        Sorting25 dataurut1 = new Sorting25(a, a.length);

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

```
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
PS D:\Semester 2\PRAKASD\PRAKTIKUM_ASD> □
```

## 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 temp=data[i];
        data[i]=data[min];
        data[min]=temp;
    }
}
```

```
public class MainSorting25 {
    public static void main(String[] args) {
        int a[]={20, 10, 2, 7, 12};
        Sorting25 dataaurut1 = new Sorting25(a, a.length);
        int b[]={30, 20, 1, 8, 14};
        Sorting25 dataaurut2 = new Sorting25(b, b.length);

        System.out.println(" == Data awal 1 == ");
        dataaurut1.tampil();
        dataaurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT
(ASC)");
        dataaurut1.tampil();

        System.out.println(" == Data awal 2 == ");
        dataaurut2.tampil();
        dataaurut2.bubbleSort();
        System.out.println("Data sudah diurutkan dengan SELECTION SORT
(ASC)");
        dataaurut2.tampil();
    }
}
```

```
== Data awal 2 ==
30 20 1 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
1 8 14 20 30
```

### c. SORTING – 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){
            j--;
        }
        data[j+1]=temp;
    }
}
```

```
public class MainSorting25 {
    public static void main(String[] args) {
        int a[]={20, 10, 2, 7, 12};
        Sorting25 dataurut1 = new Sorting25(a, a.length);
        int b[]={30, 20, 2, 8, 14};
        Sorting25 dataurut2 = new Sorting25(b, b.length);
        int c[]={40, 10, 4, 9, 3};
        Sorting25 dataurut3 = new Sorting25(c, c.length);

        System.out.println(" == Data awal 1 == ");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT
(ASC)");
        dataurut1.tampil();

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

        System.out.println(" == Data awal 3 == ");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println("Data sudah diurutkan dengan INSERTION SORT
(ASC)");
        dataurut3.tampil();
    }
}
```

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

### 6.1.2 Pertanyaan!

1. Jelaskan fungsi kode berikut

```

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

```

### **Kondisi if (data[j-1] > data[j])**

- Mengecek apakah elemen sebelumnya (data[j-1]) lebih besar dari elemen saat ini (data[j]).
- Jika iya, berarti elemen tidak dalam urutan yang benar dan perlu ditukar.

### **Penukaran nilai (swap)**

- `temp = data[j];` → Simpan nilai data[j] ke variabel temp sementara.
- `data[j] = data[j-1];` → Geser nilai data[j-1] ke data[j].
- `data[j-1] = temp;` → Masukkan kembali nilai data[j] yang sebelumnya disimpan di temp ke data[j-1].

2. Tampilkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!

```

for (int i=0; i<jumlahData-1; i++) {
    int min=i;
    for (int j=i+1; j<jumlahData; j++){
        if(data[j] < data[min]){
            min=j;
        }
    }
    int temp=data[i];
    data[i]=data[min];
    data[min]=temp;
}

```

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

```

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

```

- Perulangan `for (int i = 1; i < n; i++)`: Digunakan untuk memilih elemen yang akan disisipkan pada posisi yang tepat dalam array yang sudah terurut.
- Perulangan `while (j >= 0 && data[j] > key)`: Digunakan untuk menggeser elemen yang lebih besar ke kanan agar ada tempat untuk elemen yang sedang disisipkan (key).
-

4. . Pada Insertion sort, apakah tujuan dari perintah

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

*Untuk menyisipkan elemen ke dalam posisi yang tepat dalam bagian array yang sudah terurut.*

- *Jika elemen sebelumnya lebih besar dari elemen yang sedang disisipkan (key), maka elemen tersebut digeser ke kanan.*
- *Setelah elemen yang lebih besar bergeser, key akan ditempatkan di lokasi yang benar (data[j + 1]).*

### 6.2.1 Sorting Menggunakan Array of Object

```
public class Mahasiswa25 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa25() {

    }

    Mahasiswa25(String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }
    void tampilInformasi() {
        System.out.println("NIM: " + nim);
        System.out.println("Nama: " + nama);
        System.out.println("Kelas: " + kelas);
        System.out.println("IPK: "+ ipk);
    }
}
```

```
public class MahasiswaBerprestasi25 {
    Mahasiswa25[] listMhs= new Mahasiswa25 [5];
    int idx;

    void tambah (Mahasiswa25 m) {
        if (idx<listMhs.length){
            listMhs[idx]=m;
            idx++;
        }else {
            System.out.println("data sudah penuh");
        }
    }
    void tampil(){
        for (Mahasiswa25 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){
                    Mahasiswa25 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1]=tmp;
                }
            }
        }
    }
}
```

```

public class MahasiswaDemo25 {
    public static void main(String[] args) {
        MahasiswaBerprestasi25 list = new MahasiswaBerprestasi25();

        Mahasiswa25 m1 = new Mahasiswa25("123", "Zaidan", "2A", 3.2);
        Mahasiswa25 m2 = new Mahasiswa25("124", "Ayu", "2A", 3.5);
        Mahasiswa25 m3 = new Mahasiswa25("125", "Sofi", "2A", 3.1);
        Mahasiswa25 m4 = new Mahasiswa25("126", "Sita", "2A", 3.9);
        Mahasiswa25 m5 = new Mahasiswa25("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();
    }
}

```

```

== Data Mahasiswa sebelum sorting ==
NIM: 123
Nama: Zaidan
Kelas: 2A
IPK: 3.2

-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5

-----
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1

-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9

-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7

```



```
=====
== Data mahasiswa setelah sorting berdasarkan IPK (DESC) :
```

```
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9
```

```
=====
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7
```

```
=====
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5
```

```
=====
NIM: 123
Nama: Zaidan
Kelas: 2A
IPK: 3.2
```

```
=====
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1
```

```
=====
PS D:\Semester 2\PRAKASD\PRAKTIKUM_ASD>
```

### 6.2.2 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  $i < \text{listMhs.length} - 1$ ?  
*Karena dalam Bubble Sort melakukan perbandingan antar elemen berpasangan, dan setiap iterasi menempatkan elemen terbesar ke posisi yang benar. Oleh karena itu, kita hanya perlu melakukan iterasi hingga elemen kedua terakhir ( $\text{listMhs.length} - 1$ ), karena setelah itu elemen sudah berada di posisi yang benar.*
- B. Mengapa syarat dari perulangan j adalah  $j < \text{listMhs.length} - i$ ?  
*Karena setelah setiap iterasi, elemen terbesar akan berada di akhir array, sehingga tidak perlu membandingkannya lagi. maka, batas perulangan dalam j berkurang sebesar i setiap iterasi.*
- 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 49 kali ( $50 - 1$ ).
  - Bubble Sort memiliki 49 tahap pengurutan, karena setiap tahap akan menggelembungkan elemen terbesar ke posisi akhir yang benar.

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

```
import java.util.Scanner;
public class MahasiswaDemo25 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi25 list = new MahasiswaBerprestasi25();

        for (int i = 0; i < 5; i++) {
            System.out.println("Masukkan data 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();

            Mahasiswa25 m = new Mahasiswa25(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();

        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();
    }
}
```

Masukkan data mahasiswa ke-1:

NIM: 12234

Nama: renata

Kelas: 2A

IPK: 3,9

Masukkan data mahasiswa ke-2:

NIM: 12235

Nama: mohan

Kelas: 2A

IPK: 3,7

Masukkan data mahasiswa ke-3:

NIM: 12236

Nama: viola

Kelas: 2A

IPK: 3,5

Masukkan data mahasiswa ke-4:

NIM: 12237

Nama: wisnu

Kelas: 2A

IPK: 3,6

Masukkan data mahasiswa ke-5:

NIM: 12238

Nama: juki

Kelas: 2A

IPK: 4,0

Data mahasiswa sebelum sorting:

NIM: 12234

Nama: renata

Kelas: 2A

IPK: 3.9

NIM: 12235

Nama: mohan

Kelas: 2A

IPK: 3.7

NIM: 12236

Nama: viola

Kelas: 2A

IPK: 3.5

NIM: 12237

Nama: wisnu

Kelas: 2A

IPK: 3.6

NIM: 12238

Nama: juki

Kelas: 2A

IPK: 4.0

Data mahasiswa setelah sorting berdasarkan IPK (DESC):

NIM: 12238

Nama: juki

Kelas: 2A

IPK: 4.0

-----  
NIM: 12234

Nama: renata

Kelas: 2A

IPK: 3.9

-----  
NIM: 12235

Nama: mohan

Kelas: 2A

IPK: 3.7

-----  
NIM: 12237

Nama: wisnu

Kelas: 2A

IPK: 3.6

-----  
NIM: 12236

Nama: viola

Kelas: 2A

IPK: 3.5

-----  
Data yang sudah terurut menggunakan SELECTION SORT (ASC)

NIM: 12236

Nama: viola

Kelas: 2A

IPK: 3.5

-----  
NIM: 12237

Nama: wisnu

Kelas: 2A

IPK: 3.6

-----  
NIM: 12235

Nama: mohan

Kelas: 2A

IPK: 3.7

-----  
NIM: 12234

Nama: renata

Kelas: 2A

IPK: 3.9

-----  
NIM: 12238

Nama: juki

Kelas: 2A

IPK: 4.0

-----

Data yang sudah terurut menggunakan INSERTION SORT (ASC)

NIM: 12236  
Nama: viola  
Kelas: 2A  
IPK: 3.5

-----  
NIM: 12237  
Nama: wisnu  
Kelas: 2A  
IPK: 3.6

-----  
NIM: 12235  
Nama: mohan  
Kelas: 2A  
IPK: 3.7

-----  
NIM: 12234  
Nama: renata  
Kelas: 2A  
IPK: 3.9

-----  
NIM: 12238  
Nama: juki  
Kelas: 2A  
IPK: 4.0

-----  
PS D:\Semester 2\PRAKASD\PRAKTIKUM ASD> █

### 6.3.1 Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

```
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;
            }
        }
        Mahasiswa25 tmp= listMhs[idxMin];
        listMhs[idxMin]=listMhs[i];
        listMhs[i]=tmp;
    }
}
```

```
System.out.println("== Data yang sudah terurut menggunakan SELECTION  
SORT {ASC} ==");
list.selectionSort();
list.tampil();
}
```

```
== Data Mahasiswa sebelum sorting ==
NIM: 123
Nama: Zaidan
Kelas: 2A
IPK: 3.2

-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5

-----
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1

-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9

-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7
```

```
-----
== Data yang sudah terurut menggunakan SELECTION SORT {ASC} ==
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1

-----
NIM: 123
Nama: Zaidan
Kelas: 2A
IPK: 3.2

-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5

-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7

-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9

-----
PS D:\Semester 2\PRAKASD\PRAKTIKUM_ASD>
```

### 6.3.2 Pertanyaan

Di dalam method selection sort, terdapat baris program seperti dibawah 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!

*Untuk menemukan elemen terkecil dalam bagian array yang belum terurut*

- *Loop j mencari indeks dari elemen terkecil akan ditukar dengan nilai terkecil*
- *Setelah loop selesai, elemen tekecil akan ditukar dengan elemen pertama pada bagian belum terurut.*

#### 6.4.1 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

```
void insertionSort(){
    for (int i=1; i<listMhs.length; i++){
        Mahasiswa25 temp = listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk){
            listMhs[j]=listMhs[j-1];
        }
        listMhs[j]= temp;
    }
}
```

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

```
== Data Mahasiswa sebelum sorting ==
NIM: 123
Nama: Zaidan
Kelas: 2A
IPK: 3.2

-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5

-----
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1

-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9

-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7
```

```

=====
== Data yang sudah terurut menggunakan INSERTION SORT (ASC) ==
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1

-----
NIM: 123
Nama: Zaidan
Kelas: 2A
IPK: 3.2

-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5

-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7

-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9

=====
PS D:\Semester 2\PRAKASD\PRAKTIKUM_ASD> 

```

#### 6.4.2 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++){
        Mahasiswa25 temp = listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk < temp.ipk){
            listMhs[j]=listMhs[j-1];
        }
        listMhs[j]= temp;
    }
}

```



## 6.5 LATIHAN

```
public class Dosen25 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    Dosen25(String kd, String name, boolean jk, int usi) {
        this.kode = kd;
        this.nama = name;
        this.jenisKelamin = jk;
        this.usia = usi;
    }
    void tampil() {
        System.out.println("kode: " + kode + ", nama: " + nama + ",
jenis kelamin: " + (jenisKelamin ? "laki laki" : "perempuan") + ", usia:
" + usia);
    }
}
```

```

public class dataDosen25 {
    Dosen25[] dataDosen = new Dosen25[10];
    int idx = 0;

    void tambah(Dosen25 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println(" Data sudah penuh!");
        }
    }

    void tampil() {
        for (int i = 0; i < idx; i++) {
            dataDosen[i].tampil();
        }
    }

    void bubbleSort() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - i - 1; j++) {
                if (dataDosen[j].usia > dataDosen[j + 1].usia) {
                    Dosen25 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
    }

    void SelectionSort() {
        for (int i = 0; i < idx - 1; i++) {
            int maxIdx = i;
            for (int j = i + 1; j < idx; j++) {
                if (dataDosen[j].usia > dataDosen[maxIdx].usia) {
                    maxIdx = j;
                }
            }
            Dosen25 temp = dataDosen[maxIdx];
            dataDosen[maxIdx] = dataDosen[i];
            dataDosen[i] = temp;
        }
    }

    void insertionSort() {
        for (int i = 1; i < idx; i++) {
            Dosen25 key = dataDosen[i];
            int j = i - 1;
            while (j >= 0 && dataDosen[j].usia < key.usia) {
                dataDosen[j + 1] = dataDosen[j];
                j--;
            }
            dataDosen[j + 1] = key;
        }
    }
}

```

```

import java.util.List;
import java.util.Scanner;

public class dosenDemo25 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        dataDosen25 data = new dataDosen25();

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampil Data Dosen");
            System.out.println("3. Sorting ASC (Usia Muda ke Tua)");
            System.out.println("4. Sorting DSC (Usia Tua ke Muda)");
            System.out.println("5. Sorting DSC (Insertion Sort)");
            System.out.println("6. Keluar");
            System.out.print("Pilih: ");
            int pilih = input.nextInt();
            input.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Masukkan kode: ");
                    String kode = input.nextLine();
                    System.out.print("Masukkan nama: ");
                    String nama = input.nextLine();
                    System.out.print("Masukkan jenis kelamin (true untuk
Laki-laki, false untuk Perempuan): ");
                    boolean jk = input.nextBoolean();
                    System.out.print("Masukkan usia: ");
                    int usia = input.nextInt();
                    data.tambah(new Dosen25(kode, nama, jk, usia));
                    break;
                case 2:
                    data.tampil();
                    break;
                case 3:
                    data.bubbleSort();
                    System.out.println("Data dosen setelah sorting
(ASC).");
                    data.tampil();
                    break;
                case 4:
                    data.SelectionSort();
                    System.out.println("Data yang sudah terurut
menggunakan selection sort (DSC)");
                    data.tampil();
                    break;
                case 5:
                    data.insertionSort();
                    System.out.println("Data yang sudah terurut
menggunakan insertion sort (DSC).");
                    data.tampil();
                    break;
                case 6:
                    System.out.println("Keluar dari program.");
                    input.close();
                    return;
                default:
                    System.out.println("Pilihan tidak valid!");
            }
        }
    }
}

```

Menu:

1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Sorting DSC (Insertion Sort)
6. Keluar

Pilih: 2

kode: WH, nama: Wahyu Hardianto, jenis kelamin: laki laki, usia: 30  
kode: MM, nama: Maulana Merkuri, jenis kelamin: laki laki, usia: 32  
kode: WF, nama: Widya Fatimah, jenis kelamin: perempuan, usia: 29  
kode: NH, nama: Nur Hidayah, jenis kelamin: perempuan, usia: 35

Menu:

1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Sorting DSC (Insertion Sort)
6. Keluar

Pilih: 3

Data dosen setelah sorting (ASC).

kode: WF, nama: Widya Fatimah, jenis kelamin: perempuan, usia: 29  
kode: WH, nama: Wahyu Hardianto, jenis kelamin: laki laki, usia: 30  
kode: MM, nama: Maulana Merkuri, jenis kelamin: laki laki, usia: 32  
kode: NH, nama: Nur Hidayah, jenis kelamin: perempuan, usia: 35

Menu:

1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Sorting DSC (Insertion Sort)
6. Keluar

Pilih: 4

Data yang sudah terurut menggunakan selection sort (DSC)

kode: NH, nama: Nur Hidayah, jenis kelamin: perempuan, usia: 35  
kode: MM, nama: Maulana Merkuri, jenis kelamin: laki laki, usia: 32  
kode: WH, nama: Wahyu Hardianto, jenis kelamin: laki laki, usia: 30  
kode: WF, nama: Widya Fatimah, jenis kelamin: perempuan, usia: 29

Menu:

1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Sorting DSC (Insertion Sort)
6. Keluar

Pilih: 5

Data yang sudah terurut menggunakan insertion sort (DSC).

kode: NH, nama: Nur Hidayah, jenis kelamin: perempuan, usia: 35  
kode: MM, nama: Maulana Merkuri, jenis kelamin: laki laki, usia: 32  
kode: WH, nama: Wahyu Hardianto, jenis kelamin: laki laki, usia: 30  
kode: WF, nama: Widya Fatimah, jenis kelamin: perempuan, usia: 29

Menu:

1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Sorting DSC (Insertion Sort)
6. Keluar

Pilih: 6

Keluar dari program.

PS D:\Semester 2\PRAKASD\PRAKTIKUM\_ASD> |