

**JOB SHEET 5**  
**PRAKTIKUM ALGORITMA**  
**STRUKTUR DATA**



**Burhnauddin ihsan**  
**244107020189**  
**TI 1E/06**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2025**

## PERCOBAAN 1 Mengimplementasikan Sorting menggunakan object

### 1. Kode program

#### CLASS SORTING

```
public class Sorting06 {
    int[] data;
    int jumData;

    Sorting06(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;
        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 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;
        }
    }

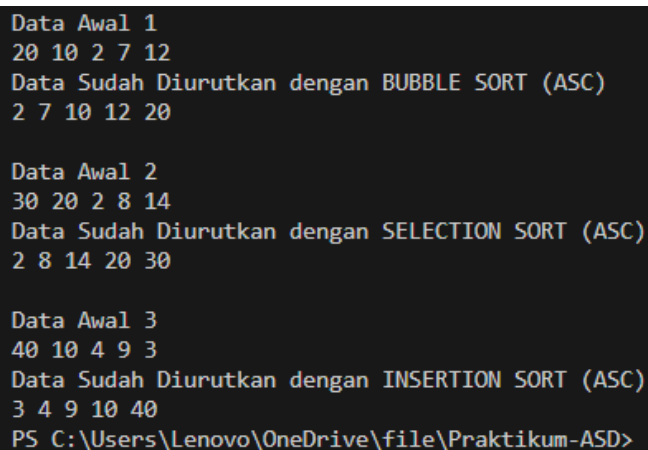
    void insertionSort() {
        for (int i = 1; i < jumData; i++) {
            int temp = data[i];
            int j = i - 1;
            while (j >= 0 && data[j] > temp) {
                data[j + 1] = data[j];
                j--;
            }
            data[j + 1] = temp;
        }
    }

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

## CLASS SORTINGMAIN

```
public class sortingMain06 {  
    public static void main(String[] args) {  
        int a[] = {20, 10, 2, 7, 12};  
        Sorting06 dataurut1 = new Sorting06(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();  
        System.out.println();  
  
        int b[] = {30, 20, 2, 8, 14};  
        Sorting06 dataurut2 = new Sorting06(b, b.length);  
        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();  
  
        int c[] = {40, 10, 4, 9, 3};  
        Sorting06 dataurut3 = new Sorting06(c, c.length);  
        System.out.println("Data Awal 3");  
        dataurut3.tampil();  
        dataurut3.insertionSort();  
        System.out.println("Data Sudah Diurutkan dengan INSERTION SORT (ASC)");  
        dataurut3.tampil();  
    }  
}
```

### 2. Hasil dari kode program



```
Data Awal 1  
20 10 2 7 12  
Data Sudah Diurutkan dengan BUBBLE SORT (ASC)  
2 7 10 12 20  
  
Data Awal 2  
30 20 2 8 14  
Data Sudah Diurutkan dengan SELECTION SORT (ASC)  
2 8 14 20 30  
  
Data Awal 3  
40 10 4 9 3  
Data Sudah Diurutkan dengan INSERTION SORT (ASC)  
3 4 9 10 40  
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

## PERTANYAAN

1. Kode ini berfungsi untuk menukar dua elemen dalam array jika mereka dalam urutan yang salah, yang merupakan langkah utama dalam algoritma Bubble Sort.

## 2. Kode program algoritma pencarian nilai minimum pada selection sort

```
(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, kondisi while ( $j \geq 0$  &&  $\text{data}[j] > \text{temp}$ ) berfungsi untuk menggeser elemen yang lebih besar ke kanan hingga menemukan posisi yang tepat bagi elemen yang sedang disisipkan. Syarat  $j \geq 0$  mencegah akses indeks negatif, sementara  $\text{data}[j] > \text{temp}$  memastikan elemen yang lebih besar digeser untuk menjaga urutan yang benar. Dengan cara ini, setiap elemen ditempatkan pada posisi yang sesuai dalam bagian array yang sudah terurut.
4. bertujuan untuk menggeser elemen yang lebih besar ke kanan agar memberi ruang bagi elemen yang sedang disisipkan.

## PERCOBAAN 2 (Sorting Menggunakan Array of Object)

### - BUBBLESORT

#### 1. Kode program

CLASS MAHASISWA

```
public class Mahasiswa06 {  
    String nim;  
    String nama;  
    String kelas;  
    double ipk;  
  
    Mahasiswa06() {  
    }  
  
    public Mahasiswa06(String nm, String name, String kls, double ip) {  
        nim = nm;  
        nama = name;  
        ipk = ip;  
        kelas = kls ;  
    }  
  
    void tampilkanInformasi(){  
        System.out.println("Nama : " + nama);  
        System.out.println("Nim : " + nim);  
        System.out.println("Kelas : " + kelas);  
        System.out.println("IPK : " + ipk);  
    }  
}
```

## CLASS MAHASISWABERPRESTASI

```
public class MahasiswaBerprestasi06 {
    Mahasiswa06[] listMhs = new Mahasiswa06[5];
    int idx;

    void tambah(Mahasiswa06 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa06 m : listMhs) {
            m.tampilkanInformasi
                ();
            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) {
                    Mahasiswa06 tmp = listMhs[j];
                    listMhs[j] = listMhs[j - 1];
                    listMhs[j - 1] = tmp;
                }
            }
        }
    }
}
```

## CLASS MAHASISWADEMO

```
public class MahasiswaDemo06 {
    public static void main(String[] args) {
        MahasiswaBerprestasi06 list = new MahasiswaBerprestasi06();
        Mahasiswa06 m1 = new Mahasiswa06("123", "Zidan", "2A", 3.2);
        Mahasiswa06 m2 = new Mahasiswa06("124", "Ayu", "2A", 3.5);
        Mahasiswa06 m3 = new Mahasiswa06("125", "Sofi", "2A", 3.1);
        Mahasiswa06 m4 = new Mahasiswa06("126", "Sita", "2A", 3.9);
        Mahasiswa06 m5 = new Mahasiswa06("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 Sesudah Sorting Berdasarkan IPK (DESC)
: ");
        list.bubbleSort();
        list.tampil();
    }
}
```

2. Hasil dari kode program

```
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 Sesudah 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\Lenovo\OneDrive\file\Praktikum-ASD>
```

## PERTANYAAN

1. A. Syarat  $i < \text{listMhs.length} - 1$ , Karena setiap iterasi akan menempatkan satu elemen terbesar ke posisi yang benar, kita hanya butuh  $n-1$  iterasi untuk mengurutkan seluruh array.

B. Syarat  $j < \text{listMhs.length} - i$ / Setelah setiap tahap, elemen terbesar sudah berada di akhir, sehingga jumlah perbandingan berkurang di iterasi berikutnya.

C. Jika  $\text{listMhs.length} = 50$ :

Perulangan  $i$  berlangsung 49 kali.

Bubble Sort menempuh 49 tahap.

Total perbandingan dalam proses: 1225 kali.

2. Modifikasi program

MAHASISWADEMO

```
import java.util.Scanner;

public class MahasiswaDemo06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi06 list = new MahasiswaBerprestasi06();

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

        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    : ");
            String ipkStr = sc.nextLine().replace(",", ".");
            double ipk = Double.parseDouble(ipkStr);

            Mahasiswa06 m = new Mahasiswa06(nim, nama, kelas, ipk);
            list.tambah(m);

            System.out.println("-----");
        }

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

## MAHASISWABERPRESTASI

```
public class MahasiswaBerprestasi06 {
    Mahasiswa06[] listMhs = new Mahasiswa06[5];
    int idx;

    void tambah(Mahasiswa06 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa06 m : listMhs) {
            if (m != null) {
                m.tampilkanInformasi();
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j] != null && listMhs[j - 1] != null) { // Cek
null untuk menghindari error
                    if (listMhs[j].ipk > listMhs[j - 1].ipk) {
                        Mahasiswa06 tmp = listMhs[j];
                        listMhs[j] = listMhs[j - 1];
                        listMhs[j - 1] = tmp;
                    }
                }
            }
        }
    }
}
```



## Hasil dari kode program

```
Data Mahasiswa Sebelum Sorting:
Masukkan Data Mahasiswa ke-1
NIM : 121343
Nama : burhan
Kelas : 1e
IPK : 3.7
-----
Masukkan Data Mahasiswa ke-2
NIM : 124431
Nama : udin
Kelas : 1e
IPK : 3.6
-----
Masukkan Data Mahasiswa ke-3
NIM : 123141
Nama : ihsan
Kelas : 1e
IPK : 3.8
-----
Masukkan Data Mahasiswa ke-4
NIM : 123341
Nama : yono
Kelas : 1e
IPK : 3.9
-----
Masukkan Data Mahasiswa ke-5
NIM : 1231332
Nama : mardi
Kelas : 1e
IPK : 3.5
-----
```

```
Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):
Nama : yono
Nim : 123341
Kelas : 1e
IPK : 3.9
Nama : ihsan
Nim : 123141
Kelas : 1e
IPK : 3.8
Nama : burhan
Nim : 121343
Kelas : 1e
IPK : 3.7
Nama : udin
Nim : 124431
Kelas : 1e
IPK : 3.6
Nama : mardi
Nim : 1231332
Kelas : 1e
IPK : 3.5
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD> |
```

## - SELECTION SORT

### 1. Kode program

#### MAHASISWABERPRESTASI

```
public class MahasiswaBerprestasi06 {
    Mahasiswa06[] listMhs = new Mahasiswa06[5];
    int idx;

    void tambah(Mahasiswa06 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa06 m : listMhs) {
            if (m != null) {
                m.tampilkanInformasi();
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j] != null && listMhs[j - 1] != null) {
                    if (listMhs[j].ipk > listMhs[j - 1].ipk) {
                        Mahasiswa06 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] != null && listMhs[idxMin] != null) {
                    if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                        idxMin = j;
                    }
                }
            }
            Mahasiswa06 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
}
```

## MAHASISWADEMO

```
import java.util.Scanner;

public class MahasiswaDemo06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi06 list = new MahasiswaBerprestasi06();

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

        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      : ");
            String ipkStr = sc.nextLine().replace(",", "."); //
            Mengantisipasi format desimal
            double ipk = Double.parseDouble(ipkStr);

            Mahasiswa06 m = new Mahasiswa06(nim, nama, kelas, ipk);
            list.tambah(m);
            System.out.println("-----");
        }

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

        System.out.println();
        System.out.println("Data Mahasiswa Setelah Sorting Menggunakan
SELECTION SORT (ASC):");
        list.selectionSort();
        list.tampil();
    }
}
```

2. Hasil dari kode program

```
Data Mahasiswa Sebelum Sorting:
Masukkan Data Mahasiswa ke-1
NIM : 123
Nama : ali
Kelas : 2b
IPK : 3.9
-----
Masukkan Data Mahasiswa ke-2
NIM : 124
Nama : ila
Kelas : 2b
IPK : 3.1
-----
Masukkan Data Mahasiswa ke-3
NIM : 125
Nama : agus
Kelas : 2b
IPK : 3.6
-----
Masukkan Data Mahasiswa ke-4
NIM : 126
Nama : tika
Kelas : 2b
IPK : 3.3
-----
Masukkan Data Mahasiswa ke-5
NIM : 127
Nama : udin
Kelas : 2b
IPK : 3.2
-----
```

```
Data Mahasiswa Setelah Sorting Menggunakan SELECTION SORT (ASC):
Nama : ila
Nim : 124
Kelas : 2b
IPK : 3.1
Nama : udin
Nim : 127
Kelas : 2b
IPK : 3.2
Nama : tika
Nim : 126
Kelas : 2b
IPK : 3.3
Nama : agus
Nim : 125
Kelas : 2b
IPK : 3.6
Nama : ali
Nim : 123
Kelas : 2b
IPK : 3.9
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

## PERTANYAAN

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

Kode tersebut merupakan bagian dari algoritma Selection Sort yang berfungsi untuk mencari elemen dengan IPK terkecil dalam sisa array listMhs. Variabel idxMin menyimpan indeks elemen dengan IPK terkecil yang ditemukan sejauh ini, dimulai dari indeks i. Perulangan for dengan variabel j akan menelusuri elemen-elemen setelah i untuk membandingkan nilai IPK. Jika ditemukan IPK yang lebih kecil, indeks idxMin diperbarui. Setelah proses ini selesai, elemen dengan IPK terkecil akan ditukar dengan elemen di indeks i, sehingga memastikan data terurut dari yang terkecil ke terbesar.

### - INSERTION SORT

#### 1. Kode program

```
public class MahasiswaBerprestasi06 {
    Mahasiswa06[] listMhs = new Mahasiswa06[5];
    int idx;

    void tambah(Mahasiswa06 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa06 m : listMhs) {
            if (m != null) {
                m.tampilkanInformasi();
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j] != null && listMhs[j - 1] != null) {
                    if (listMhs[j].ipk > listMhs[j - 1].ipk) {
                        Mahasiswa06 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] != null && listMhs[idxMin] != null) {
                if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                    idxMin = j;
                }
            }
        }
        Mahasiswa06 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa06 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1] != null && temp != null &&
listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
}

```

## 2. hasil dari kode program

```

Storage (Path: C:\Users\user\AppData\Local\Temp\10147...
Data Mahasiswa Sebelum Sorting:
Masukkan Data Mahasiswa ke-1
NIM    : 111
Nama   : ayu
Kelas : 2c
IPK    : 3.7
-----
Masukkan Data Mahasiswa ke-2
NIM    : 222
Nama   : dika
Kelas : 2c
IPK    : 3.0
-----
Masukkan Data Mahasiswa ke-3
NIM    : 333
Nama   : ila
Kelas : 2c
IPK    : 3.8
-----
Masukkan Data Mahasiswa ke-4
NIM    : 444
Nama   : susi
Kelas : 2c
IPK    : 3.1
-----
Masukkan Data Mahasiswa ke-5
NIM    : 555
Nama   : yayuk
Kelas : 2c
IPK    : 3.4
-----

```

```
Data Yang Sudah Terurut Menggunakan INSERTION SORT (ASC)
NIM   : 222
Nama  : dika
Kelas : 2c
IPK   : 3.0
-----
NIM   : 444
Nama  : susi
Kelas : 2c
IPK   : 3.1
-----
NIM   : 555
Nama  : yayuk
Kelas : 2c
IPK   : 3.4
-----
NIM   : 111
Nama  : ayu
Kelas : 2c
IPK   : 3.7
-----
NIM   : 333
Nama  : ila
Kelas : 2c
IPK   : 3.8
-----
```

## PERTANYAAN

Mengubah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending, Perubahan yang dilakukan adalah pada fungsi insertionSort() pada bagian while Tanda perbandingan diubah dari > menjadi <:

Kode program:

```
void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa06 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) { // Perbaikan
            indeks dari j1 ke j - 1
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Data Mahasiswa Sebelum Sorting :

Masukkan Data Mahasiswa ke-1

NIM : 111

Nama : ayu

Kelas : 2c

IPK : 3.7

-----  
Masukkan Data Mahasiswa ke-2

NIM : 222

Nama : dika

Kelas : 2c

IPK : 3.0

-----  
Masukkan Data Mahasiswa ke-3

NIM : 333

Nama : ila

Kelas : 2c

IPK : 3.8

-----  
Masukkan Data Mahasiswa ke-4

NIM : 444

Nama : yayuk

Kelas : 2c

IPK : 3.1

-----  
Masukkan Data Mahasiswa ke-5

NIM : 555

Nama : susi

Kelas : 2c

IPK : 3.4

Data Yang Sudah Terurut Menggunakan INSERTION SORT (DESC)

NIM : 333

Nama : ila

Kelas : 2c

IPK : 3.8

-----  
NIM : 111

Nama : ayu

Kelas : 2c

IPK : 3.7

-----  
NIM : 555

Nama : susi

Kelas : 2c

IPK : 3.4

-----  
NIM : 444

Nama : yayuk

Kelas : 2c

IPK : 3.1

-----  
NIM : 222

Nama : dika

Kelas : 2c

IPK : 3.0



## LATIHAN

### 1. Kode program

#### DOSEN

```
public class Dosen06 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    public Dosen06(String kd, String name, boolean jk, int age) {
        this.kode = kd;
        this.nama = name;
        this.jenisKelamin = jk;
        this.usia = age;
    }

    public void tampil() {
        System.out.println();
        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("-----");
    }
}
```

#### DATADOSEN

```
public class DataDosen06 {
    Dosen06[] dataDosen = new Dosen06[10];
    int idx = 0;

    void tambah(Dosen06 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data Dosen sudah penuh!");
        }
    }

    void tampil() {
        if (idx == 0) {
            System.out.println("Belum ada data.");
        } else {
            for (int i = 0; i < idx; i++) {
                dataDosen[i].tampil();
            }
        }
    }
}
```

```

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) {
                Dosen06 temp = dataDosen[j];
                dataDosen[j] = dataDosen[j + 1];
                dataDosen[j + 1] = temp;
            }
        }
    }
}

void sortingDSC() {
    for (int i = 0; i < idx - 1; i++) {
        int maxIndex = i;
        for (int j = i + 1; j < idx; j++) {
            if (dataDosen[j].usia > dataDosen[maxIndex].usia) {
                maxIndex = j;
            }
        }
        Dosen06 temp = dataDosen[maxIndex];
        dataDosen[maxIndex] = dataDosen[i];
        dataDosen[i] = temp;
    }
}

void insertionSort() {
    for (int i = 1; i < idx; i++) {
        Dosen06 temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
}
}

```

## DOSENMAIN

```

import java.util.Scanner;

public class Dosenmain06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen06 listDosen = new DataDosen06();

        while (true) {
            System.out.println("===== Menu =====");
            System.out.println("1. Tambahkan Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Urutkan Data Bubble Sort (ASC)");
            System.out.println("4. Urutkan Data Selection Sort (DSC)");
            System.out.println("5. Urutkan Data Insertion Sort (DSC)");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

```

```

switch (pilihan) {
    case 1:
        System.out.println();
        for (int i = 0; i < 10; i++) {
            System.out.print("Kode : ");
            String kode = sc.nextLine();
            System.out.print("Nama : ");
            String nama = sc.nextLine();
            System.out.print("Jenis Kelamin (true = Laki-laki, false
= Perempuan) : ");

            boolean jk = sc.nextBoolean();
            System.out.print("Usia : ");
            int usia = sc.nextInt();
            sc.nextLine();
            System.out.println();
            Dosen06 dsn = new Dosen06(kode, nama, jk, usia);
            listDosen.tambah(dsn);
        }
        break;
    case 2:
        System.out.println();
        System.out.println("=== Data Keseluruhan ===");
        listDosen.tampil();
        break;
    case 3:
        System.out.println();
        System.out.println("Data telah diurutkan (ASC - Bubble
Sort).");

        listDosen.sortingASC();
        listDosen.tampil();
        break;
    case 4:
        System.out.println();
        System.out.println("Data telah diurutkan (DSC - Selection
Sort).");

        listDosen.sortingDSC();
        listDosen.tampil();
        break;
    case 5:
        System.out.println();
        System.out.println("Data telah diurutkan (DSC - Insertion
Sort).");

        listDosen.insertionSort();
        listDosen.tampil();
        break;
    case 0:
        System.out.println("Keluar dari program.");
        sc.close();
        return;
    default:
        System.out.println("Pilihan tidak valid.");
}
}
}
}

```

## 2. Hasil dari kode program

```
java -gc_aws -i Praktikum_ASD_3170402021 - DosenRaditoo
===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 1

Kode : 123
Nama : udin
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 45

Kode : 124
Nama : burhan
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 34

Kode : 125
Nama : pandu
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 65

Kode : 126
Nama : kesia
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 34

Kode : 127
Nama : hanif
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 32

Kode : 128
Nama : desta
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 62

Kode : 129
Nama : aqila
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 31

Kode : 130
Nama : marsya
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 67

Kode : 131
Nama : rizky
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 21

Kode : 132
Nama : radit
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 39
```

```
===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 2
```

```
=== Data Keseluruhan ===
```

```
Kode : 123
Nama : udin
Jenis Kelamin : Laki-laki
Usia : 45
-----
```

```
Kode : 124
Nama : burhan
Jenis Kelamin : Laki-laki
Usia : 34
-----
```

```
Kode : 125
Nama : pandu
Jenis Kelamin : Laki-laki
Usia : 65
-----
```

```
Kode : 126
Nama : kesia
Jenis Kelamin : Perempuan
Usia : 34
-----
```

```
Kode : 127
Nama : hanif
Jenis Kelamin : Laki-laki
Usia : 32
-----
```

```
Kode : 128
Nama : desta
Jenis Kelamin : Laki-laki
Usia : 62
-----
```

```
Kode : 129
Nama : aqila
Jenis Kelamin : Perempuan
Usia : 31
-----
```

```
Kode : 130
Nama : marsya
Jenis Kelamin : Perempuan
Usia : 67
-----
```

```
Kode : 131
Nama : rizky
Jenis Kelamin : Laki-laki
Usia : 21
-----
```

```
Kode : 132
Nama : radit
Jenis Kelamin : Laki-laki
Usia : 39
-----
```

===== Menu =====

1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar

Pilih menu: 3

Data telah diurutkan (ASC - Bubble Sort).

Kode : 131

Nama : rizky

Jenis Kelamin : Laki-laki

Usia : 21

-----

Kode : 129

Nama : aqila

Jenis Kelamin : Perempuan

Usia : 31

-----

Kode : 127

Nama : hanif

Jenis Kelamin : Laki-laki

Usia : 32

-----

Kode : 124

Nama : burhan

Jenis Kelamin : Laki-laki

Usia : 34

-----

Kode : 126

Nama : kesia

Jenis Kelamin : Perempuan

Usia : 34

-----

Kode : 132

Nama : radit

Jenis Kelamin : Laki-laki

Usia : 39

-----

Kode : 123

Nama : udin

Jenis Kelamin : Laki-laki

Usia : 45

-----

Kode : 128

Nama : desta

Jenis Kelamin : Laki-laki

Usia : 62

-----

Kode : 125

Nama : pandu

Jenis Kelamin : Laki-laki

Usia : 65

-----

Kode : 130

Nama : marsya

Jenis Kelamin : Perempuan

Usia : 67

-----

===== Menu =====

1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar

Pilih menu: 4

Data telah diurutkan (DSC - Selection Sort).

Kode : 130  
Nama : marsya  
Jenis Kelamin : Perempuan  
Usia : 67  
-----

Kode : 125  
Nama : pandu  
Jenis Kelamin : Laki-laki  
Usia : 65  
-----

Kode : 128  
Nama : desta  
Jenis Kelamin : Laki-laki  
Usia : 62  
-----

Kode : 123  
Nama : udin  
Jenis Kelamin : Laki-laki  
Usia : 45  
-----

Kode : 132  
Nama : radit  
Jenis Kelamin : Laki-laki  
Usia : 39  
-----

Kode : 126  
Nama : kesia  
Jenis Kelamin : Perempuan  
Usia : 34  
-----

Kode : 124  
Nama : burhan  
Jenis Kelamin : Laki-laki  
Usia : 34  
-----

Kode : 127  
Nama : hanif  
Jenis Kelamin : Laki-laki  
Usia : 32  
-----

Kode : 129  
Nama : aqila  
Jenis Kelamin : Perempuan  
Usia : 31  
-----

Kode : 131  
Nama : rizky  
Jenis Kelamin : Laki-laki  
Usia : 21  
-----

```
-----  
===== Menu =====  
1. Tambahkan Data Dosen  
2. Tampilkan Data Dosen  
3. Urutkan Data Bubble Sort (ASC)  
4. Urutkan Data Selection Sort (DSC)  
5. Urutkan Data Insertion Sort (DSC)  
0. Keluar  
Pilih menu: 5
```

Data telah diurutkan (DSC - Insertion Sort).

```
Kode : 130  
Nama : marsya  
Jenis Kelamin : Perempuan  
Usia : 67  
-----
```

```
Kode : 125  
Nama : pandu  
Jenis Kelamin : Laki-laki  
Usia : 65  
-----
```

```
Kode : 128  
Nama : desta  
Jenis Kelamin : Laki-laki  
Usia : 62  
-----
```

```
Kode : 123  
Nama : udin  
Jenis Kelamin : Laki-laki  
Usia : 45  
-----
```

```
Kode : 132  
Nama : radit  
Jenis Kelamin : Laki-laki  
Usia : 39  
-----
```

```
Kode : 126  
Nama : kesia  
Jenis Kelamin : Perempuan  
Usia : 34  
-----
```

```
Kode : 124  
Nama : burhan  
Jenis Kelamin : Laki-laki  
Usia : 34  
-----
```

```
Kode : 127  
Nama : hanif  
Jenis Kelamin : Laki-laki  
Usia : 32  
-----
```

```
Kode : 129  
Nama : aqila  
Jenis Kelamin : Perempuan  
Usia : 31  
-----
```

```
Kode : 131  
Nama : rizky  
Jenis Kelamin : Laki-laki  
Usia : 21  
-----
```

```
-----  
===== Menu =====  
1. Tambahkan Data Dosen  
2. Tampilkan Data Dosen  
3. Urutkan Data Bubble Sort (ASC)  
4. Urutkan Data Selection Sort (DSC)  
5. Urutkan Data Insertion Sort (DSC)  
0. Keluar  
Pilih menu: 0  
Keluar dari program.  
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```