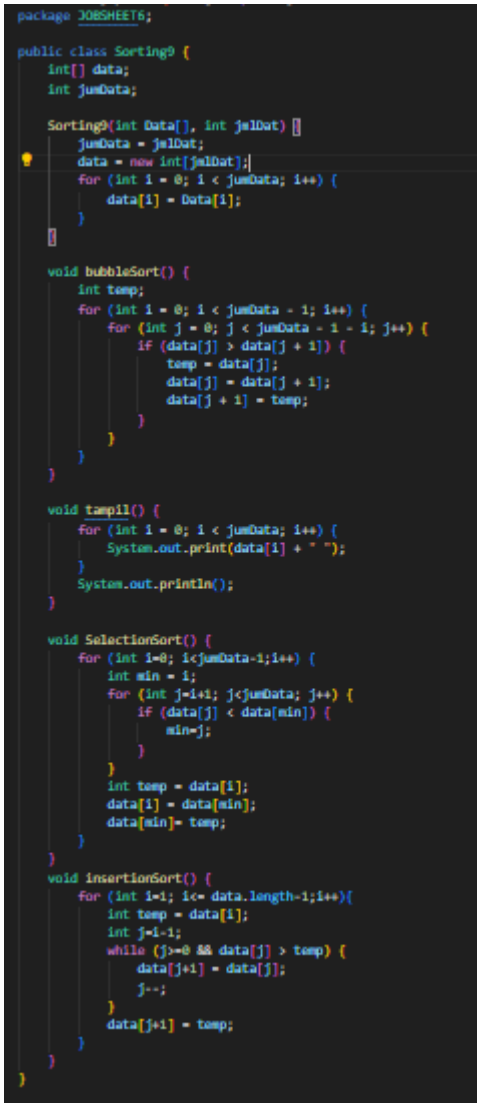
Jobsheet 6

Praktikum 1

Method Class



```
package JOBSHEET6;

public class Sorting9 {
    int[] data;
    int jumData;

    Sorting9(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 = 0; j < jumData - 1 - i; j++) {
                if (data[j] > data[j + 1]) {
                    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();
    }

    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<= 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;
        }
    }
}
```

Main Class

```
public class SortingMain9 {
    public static void main(String[] args) {

        System.out.println(x:"Data Awal:");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println(x:"Data setelah diurutkan dengan Bubble Sort (ASC):");
        dataurut1.tampil();

        int b[] = {30,20,2,8,14};
        Sorting9 dataurut2 = new Sorting9(b, b.length);

        System.out.println(x:"Data Awal 2:");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println(x:"Data setelah diurutkan dengan Selection Sort (ASC):");
        dataurut2.tampil();

        int c[]= {40,10,4,9,3};
        Sorting9 dataurut3 = new Sorting9(c, c.length);
        System.out.println(x:"Data Awal 3:");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println(x:"Data setelah diurutkan dengan Insertion Sort (ASC):");
        dataurut3.tampil();


    }
}
```

RUN PROGRAM

```
Data Awal:
20 10 2 7 12
Data setelah diurutkan dengan Bubble Sort (ASC):
2 7 10 12 20
Data Awal 2:
30 20 2 8 14
Data setelah diurutkan dengan Selection Sort (ASC):
2 8 14 20 30
Data Awal 3:
40 10 4 9 3
Data setelah diurutkan dengan Insertion Sort (ASC):
3 4 9 10 40
Data setelah diurutkan dengan Insertion Sort (ASC):
3 4 9 10 40
PS C:\Users\PC\Praktikum-ASD-1>
```

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;
}
```

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!
3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan `while (j>=0 && data[j]>temp)`
4. Pada Insertion sort, apakah tujuan dari perintah `data[j+1]= data[j];`

JAWABAN

1. Jika elemen sebelumnya (data[j-1]) lebih besar dari elemen saat ini (data[j]), maka kedua elemen tersebut ditukar Tujuannya adalah untuk menggeser elemen yang lebih besar ke posisi yang lebih tinggi dalam array, sehingga elemen terbesar akan berada di akhir setelah beberapa iterasi.
2.
```
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;  // Menyimpan indeks elemen terkecil

    }

  }

  int temp = data[i];

     data[i] = data[min];
     data[min] = temp;  // Menukar elemen terkecil dengan elemen di posisi awal
  }
```

3. **j >= 0** → Perulangan hanya berjalan jika j masih dalam batas array (tidak keluar dari indeks negative

   data[j] > temp → Jika elemen saat ini lebih besar dari elemen yang sedang disisipkan (temp), maka elemen akan digeser ke kanan untuk memberi ruang bagi elemen yang lebih kecil.

4. Tujuan dari perintah ini adalah **menggeser elemen ke kanan** untuk memberi ruang bagi elemen yang lebih kecil (`temp`).

   Ketika menemukan elemen yang lebih besar dari `temp`, elemen tersebut dipindahkan satu posisi ke kanan (`j+1`).

   Hal ini dilakukan berulang kali sampai menemukan posisi yang benar untuk menyisipkan elemen (`temp`).

Praktikum 2

BUBBLESORT

CLASS MAHASISWA BERPRESTASI

```
package JOBSHEET6;

public class MahasiswaBerprestasi9 {
    Mahasiswa9[] listMhs = new Mahasiswa9[5];
    int idx;

    void tambah(Mahasiswa9 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println(x:"Data Sudah Penuh");
        }
    }

    void tampil() {
        for (Mahasiswa9 m : listMhs) {
            if (m != null) {
                m.tampilInformasi();
                System.out.println(x:"----------------------------");
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 0; j < listMhs.length - 1 - i; j++) {
                if (listMhs[j].ipk < listMhs[j + 1].ipk) {
                    Mahasiswa9 tmp = listMhs[j];
                    listMhs[j] = listMhs[j + 1];
                    listMhs[j + 1] = tmp;
                }
            }
        }
    }
```

CLASS MAHSISWA NO.ABSENSI

```java
package JOBSHEET6;

public class Mahasiswa9 {
    String nim, nama, kelas;
    double ipk;

    Mahasiswa9() {
    }

    Mahasiswa9(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);
    }
}
```

CLASS MAHSISWA DEMO

```java
package JOBSHEET6;

public class MahasiswaDemo9 {
    Run | Debug
    public static void main(String[] args) {
        MahasiswaBerprestasi9 list = new MahasiswaBerprestasi9();
        Mahasiswa9 m1 = new Mahasiswa9(nm:"123", name:"Zidan", kls:"2A", ip:3.2);
        Mahasiswa9 m2 = new Mahasiswa9(nm:"124", name:"Ayu", kls:"2A", ip:3.5);
        Mahasiswa9 m3 = new Mahasiswa9(nm:"125", name:"Agus", kls:"2A", ip:3.1);
        Mahasiswa9 m4 = new Mahasiswa9(nm:"126", name:"Tika", kls:"2A", ip:3.9);
        Mahasiswa9 m5 = new Mahasiswa9(nm:"127", name:"Udin", kls:"2A", ip:3.7);


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

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

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

RUN PROGRAM

```
Data Mahasiswa Sebelum Sorting:
Nama: Ali
NIM:128
Kelas: 2B
IPK: 3.9
----------------------------
Nama: Ila
NIM:129
Kelas: 2B
IPK: 3.1
----------------------------
Nama: Agus
NIM:130
Kelas: 2B
IPK: 3.6
----------------------------
Nama: Tika
NIM:131
Kelas: 2B
IPK: 3.3
----------------------------
Nama: Udin
NIM:132
Kelas: 2B
IPK: 3.2
----------------------------
Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):
Nama: Ali
NIM:128
Kelas: 2B
IPK: 3.9
----------------------------
Nama: Agus
NIM:130
Kelas: 2B
IPK: 3.6
----------------------------
Nama: Tika
NIM:131
Kelas: 2B
IPK: 3.3
----------------------------
Nama: Udin
NIM:132
Kelas: 2B
IPK: 3.2
----------------------------
Nama: Ila
NIM:129
Kelas: 2B
```

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<listMhs.length-1 ?

b. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?

c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?

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

JAWABAN

1.

a. `i< listMhs.length - 1` digunakan karena pada bubble sort, setiap iterasi akan menempatkan satu elemen terbesar ke posisi akhir array. Karena elemen terakhir secara otomatis sudah berada di posisi yang benar setelah `listMhs.length - 1` iterasi, maka tidak perlu melakukan iterasi penuh hingga panjang array, cukup sampai indeks ke `listMhs.length - 1`.

b. Syarat `j < listMhs.length - 1 - i` digunakan untuk menghindari perbandingan yang tidak perlu.

c. Banyaknya perulangan `i` akan terjadi sebanyak `50 - 1 = 49` kali.

2.

```
package JOBSHEET6;

import java.util.Scanner;

public class Mahasiswa9 {
    String nim, nama, kelas;
    double ipk;

    Mahasiswa9() {
    }

    Mahasiswa9(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

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

SELECTION SORT

Class mahsiswa berprestasi

MENAMBAHKAN VOID 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;
            }
        }

        Mahasiswa9 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

Class Mahasiswa Demo

```
package JOBSHEET6;

public class MahasiswaDemo9 {
    Run | Debug
    public static void main(String[] args) {
        MahasiswaBerprestasi9 list = new MahasiswaBerprestasi9();
        Mahasiswa9 m1 = new Mahasiswa9(nim:"123", nama:"Ali", kelas:"2A", ipk:3.9);
        Mahasiswa9 m2 = new Mahasiswa9(nim:"124", nama:"Ila", kelas:"2A", ipk:3.1);
        Mahasiswa9 m3 = new Mahasiswa9(nim:"125", nama:"Agus", kelas:"2A", ipk:3.6);
        Mahasiswa9 m4 = new Mahasiswa9(nim:"126", nama:"Tika", kelas:"2A", ipk:3.3);
        Mahasiswa9 m5 = new Mahasiswa9(nim:"127", nama:"Udin", kelas:"2A", ipk:3.2);


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

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

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

        System.out.println(x:"Data Yang Sudah Terurut Menggunakan Selection Sort (ASC):");
        list.selectionSort();
        list.tampil();
```

Run Program

```
Data Yang Sudah Terurut Menggunakan Selection Sort (ASC):
Nama: Ila
NIM: 124
Kelas: 2A
IPK: 3.1
----------------------------
Nama: Udin
NIM: 127
Kelas: 2A
IPK: 3.2
----------------------------
Nama: Tika
NIM: 126
Kelas: 2A
IPK: 3.3
----------------------------
Nama: Agus
NIM: 125
Kelas: 2A
IPK: 3.6
----------------------------
Nama: Ali
NIM: 123
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!

Jawaban

Untuk menemukan elemen dengan nilai terkecil dalam sisa array untuk setiap iterasi, sehingga proses sorting dapat berjalan dengan benar. Inisialisasi idxMin = i → Menandai indeks elemen terkecil saat ini.

Perulangan j = i + 1 hingga akhir array → Mencari elemen terkecil berikutnya di sisa array.Kondisi if (listMhs[j].ipk < listMhs[idxMin].ipk) Jika ditemukan elemen dengan nilai lebih kecil dari yang saat ini dianggap minimum, Maka indeks idxMin diperbarui ke indeks baru (j).

INSERTION SORT

Class Mahasiswa Berprestasi

Tambahkan kode ini

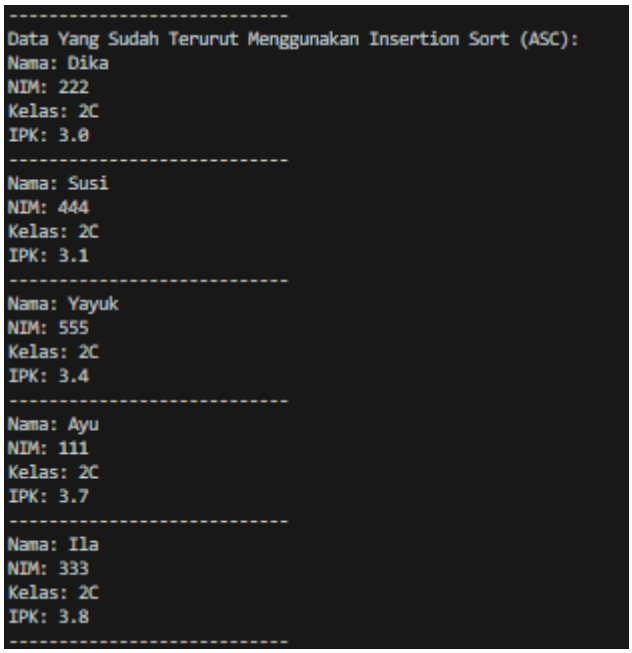
```
void insetionSort(){
    for(int i=1; i<listMhs.length; i++){
        Mahasiswa9 temp =listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk){
            listMhs[j]=listMhs[j-1];
            j--;
        }
        listMhs[j]=temp;
    }
}
```

Class Mahasiswa Demo

Tambahkan List tampil Insertion Sort di class tersebut

```
public static void main(String[] args) {
    MahasiswaBerprestasi9 list = new MahasiswaBerprestasi9();
    Mahasiswa9 m1 = new Mahasiswa9(nim:"111", nama:"Ayu", kelas:"2C", ipk:3.7);
    Mahasiswa9 m2 = new Mahasiswa9(nim:"222", nama:"Dika", kelas:"2C", ipk:3.0);
    Mahasiswa9 m3 = new Mahasiswa9(nim:"333", nama:"Ila", kelas:"2C", ipk:3.8);
    Mahasiswa9 m4 = new Mahasiswa9(nim:"444", nama:"Susi", kelas:"2C", ipk:3.1);
    Mahasiswa9 m5 = new Mahasiswa9(nim:"555", nama:"Yayuk", kelas:"2C", ipk:3.4);


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

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

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

    System.out.println(x:"Data Yang Sudah Terurut Menggunakan Selection Sort (ASC):");
    list.selectionSort();
    list.tampil();

    System.out.println(x:"Data Yang Sudah Terurut Menggunakan Insertion Sort (ASC):");
    list.insetionSort();
    list.tampil();
```

Run Program



```
----------------------------
Data Yang Sudah Terurut Menggunakan Insertion Sort (ASC):
Nama: Dika
NIM: 222
Kelas: 2C
IPK: 3.0
----------------------------
Nama: Susi
NIM: 444
Kelas: 2C
IPK: 3.1
----------------------------
Nama: Yayuk
NIM: 555
Kelas: 2C
IPK: 3.4
----------------------------
Nama: Ayu
NIM: 111
Kelas: 2C
IPK: 3.7
----------------------------
Nama: Ila
NIM: 333
Kelas: 2C
IPK: 3.8
----------------------------
```

Pertanyaan

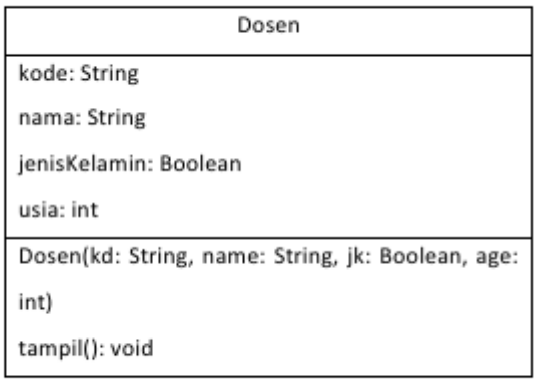
Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.

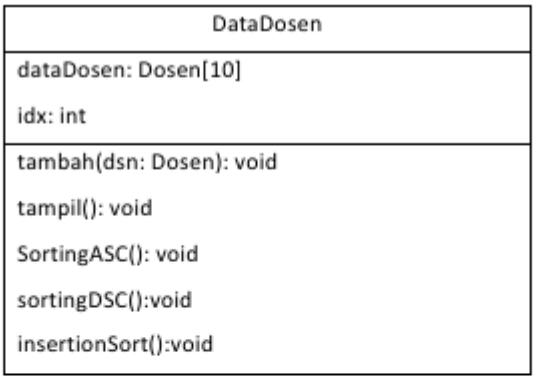
Jawaban

```
    }
    void insertionSort() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa9 temp = listMhs[i];
            int j = i;

            while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
                listMhs[j] = listMhs[j - 1];
                j--;
            }
            listMhs[j] = temp;
        }
    }

}
```

Latihan Praktikum

Perhatikan class diagram dibawah ini:





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 bublle 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.

Class Dosen

```java
package JOBSHEET6;
public class Dosen {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

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

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

Class DataDosen

```
package JOBSHEET6;
public class DataDosen {
    Dosen[] dataDosen = new Dosen[10];
    int idx = 0;

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

    public void tampil() {
        if (idx == 0) {
            System.out.println(x:"Data dosen kosong.");
            return;
        }

        for (int i = 0; i < idx; i++) {
            System.out.println("Data Dosen ke-" + (i + 1) + ":");
            dataDosen[i].tampil();
            System.out.println();
        }
    }

    public void sortingASC() {
        // Bubble Sort
        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) {
                    // Tukar data
                    Dosen temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
        tampil();
    }

    public void sortingDSC() {
        // Insertion Sort
        for (int i = 1; i < idx; i++) {
            Dosen temp = dataDosen[i];
            int j = i - 1;

            while (j >= 0 && dataDosen[j].usia < temp.usia) {
                dataDosen[j + 1] = dataDosen[j];
                j = j - 1;
            }
            dataDosen[j + 1] = temp;
        }
        tampil();
    }
}
```

Class DosenMain

```java
package JOBSHEET6;
import java.util.Scanner;

public class DosenMain {
    Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        DataDosen data = new DataDosen();
        int pilihan;

        do {
            System.out.println(x:"Menu:");
            System.out.println(x:"1. Tambah Data Dosen");
            System.out.println(x:"2. Tampil Data Dosen");
            System.out.println(x:"3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua");
            System.out.println(x:"4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda");
            System.out.println(x:"0. Keluar");
            System.out.print(s:"Pilihan: ");
            pilihan = input.nextInt();
            input.nextLine();
            switch (pilihan) {
                case 1:
                    System.out.print(s:"Kode: ");
                    String kode = input.nextLine();
                    System.out.print(s:"Nama: ");
                    String nama = input.nextLine();
                    System.out.print(s:"Jenis Kelamin (L/P): ");
                    boolean jk = input.nextLine().equalsIgnoreCase(anotherString:"L");
                    System.out.print(s:"Usia: ");
                    int usia = input.nextInt();
                    input.nextLine();

                    Dosen dsn = new Dosen(kode, nama, jk, usia);
                    data.tambah(dsn);
                    System.out.println(x:"Data dosen berhasil ditambahkan.");
                    break;
                case 2:
                    data.tampil();
                    break;
                case 3:
                    data.sortingASC();
                    break;
                case 4:
                    data.sortingDSC();
                    break;
                case 0:
                    System.out.println(x:"Program selesai.");
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid.");
            }
        } while (pilihan != 0);
    }
}
```

Run Program

```
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
0. Keluar
Pilihan: 1
Kode: 1
Nama: KOKO
Jenis Kelamin (L/P): L
Usia: 20
Data dosen berhasil ditambahkan.
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
0. Keluar
Pilihan: 1
Kode: 3
Nama: KOLO
Jenis Kelamin (L/P): P
Usia: 28
Data dosen berhasil ditambahkan.
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
0. Keluar
Pilihan: 1
Kode: 7
Nama: MUANI
Jenis Kelamin (L/P): L
Usia: 80
Data dosen berhasil ditambahkan.
```

```
Pilihan: 2
Data Dosen ke-1:
Kode: 1
Nama: KOKO
Jenis Kelamin: Laki-laki
Usia: 20

Data Dosen ke-2:
Kode: 3
Nama: KOLO
Jenis Kelamin: Perempuan
Usia: 28

Data Dosen ke-3:
Kode: 7
Nama: MUANI
Jenis Kelamin: Laki-laki
Usia: 80

Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
0. Keluar
Pilihan: 3
Data Dosen ke-1:
Kode: 1
Nama: KOKO
Jenis Kelamin: Laki-laki
Usia: 20

Data Dosen ke-2:
Kode: 3
Nama: KOLO
Jenis Kelamin: Perempuan
Usia: 28

Data Dosen ke-3:
Kode: 7
Nama: MUANI
Jenis Kelamin: Laki-laki
Usia: 80
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
0. Keluar
Pilihan: 4
Data Dosen ke-1:
Kode: 7
Nama: MUANI
Jenis Kelamin: Laki-laki
Usia: 80

Data Dosen ke-2:
Kode: 3
Nama: KOLO
Jenis Kelamin: Perempuan
Usia: 28

Data Dosen ke-3:
Kode: 1
Nama: KOKO
Jenis Kelamin: Laki-laki
Usia: 20

Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
0. Keluar
Pilihan: 0
Program selesai.
```