Laporan Algoritma Dan Struktur Data

Jobsheet 6

Ghazwan Ababil

244107020151

1E – Teknik Informatika

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

1. Praktikum

1.1 Percobaan 1: Mengimplementasikan Sorting Menggunakan Object

1.1.1 Kode Program

a. SORTING – BUBBLE SORT

1. Membuat package Jobsheet5, lalu membuat class, konstruktor berparameter, beserta atribut dan method bubbleShort untuk mengurutkan data array menggunakan algoritma Bubble Sort serta method tampil untuk menampilkan data pada class Sorting11

```
package Jobsheet6;

public class Sorting11 {
    int[] data;
    int jumData;

    Sorting11(int Data[], int jmlDat) {
        jumData = jmlDat;
        data = new int[jumData];
        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 < data.length; i++) {
            System.out.print(data[i] + " ");
        }
        System.out.println();
    }
}
```

2. Membuat Class untuk mengeksekusi dengan nama SortingMain11, kemudian mendeklarasikan array dengan nama a[] lalu isi array tersebut. Setelah itu menginstansiasi objek baru dari class Sorting11 dan disertai dengan

parameter lalu melakukan pemanggilan method bubbleSort dan tampil pada objek tersebut.

```
package Jobsheet6;

public class SortingMain11 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        Sorting11 dataurut1 = new Sorting11(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();
    }
}
```

3. Hasil Running Kode Program

```
Data awal 1:
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
```

b. SORTING – SELECTION SORT

1. Menambahkan method untuk mengurutkan data menggunakan algoritma Selection Sort dengan nama SelectionSort pada class Sorting11

```
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;
    }
}
```

2. Mendeklarasikan array baru dengan nama b[] lalu isi array tersebut. Setelah itu menginstansiasi objek baru dari class Sorting11 dan disertai dengan parameter lalu melakukan pemanggilan method SelectionSort dan tampil pada objek tersebut.

```
int b[] = {30,20,2,8,14};
Sorting11 dataurut2 = new Sorting11(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();
```

3. Hasil Running 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
```

c. SORTING – INSERTION SORT

1. Menambahkan method untuk mengurutkan data menggunakan algoritma Insertion Sort dengan nama insertionSort pada class Sorting11

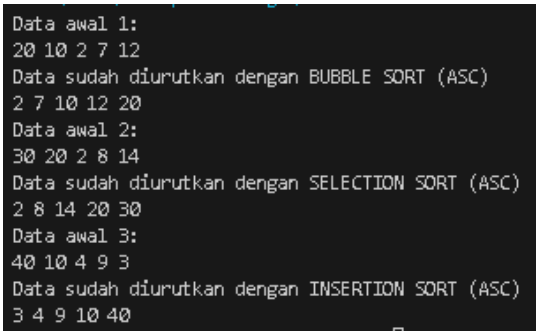
```
void insertionSort(){
    for (int i = 0; i < data.length-1; i++) {
        int temp = data[i];
        int j = i-1;
        while (j >= 0 && data[j] > temp) {
            data[j+1] = data[j];
            j--;
        }
        data[j+1] = temp;
    }
}
```

2. Mendeklarasikan array baru dengan nama c[] lalu isi array tersebut. Setelah itu menginstansiasi objek baru dari class Sorting11 dan disertai dengan parameter lalu melakukan pemanggilan method insertionSort dan tampil pada objek tersebut.

```
int c[] = {40,10,4,9,3};
Sorting11 dataurut3 = new Sorting11(c, c.length);

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

3. Hasil Running Kode Program



1.1.2 Pertanyaan

1) Kode program tersebut if tersebut berfungsi untuk melakukan pengecekan nilai elemen pada indeks array kiri (0/ j-1) apakah lebih besar dari elemen pada indeks kanan (1 / j), jika kondisi terpenuhi atau benar maka elemen pada indeks array kiri akan bertukar tempat dengan elemen pada indeks array kanan dengan perantara variabel temp untuk menyimpan sementara nilai elemen pada indeks array kanan.
2) Kode program yang merupakan algoritma untuk melakukan pencarian nilai umum pada selection sort adalah

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

Kode tersebut memiliki proses pertama kali menyimpan indeks elemen pertama pada array sebagai nilai minimum sementara, kemudian dilakukan perulangan sebanyak panjang array untuk mengecek apakah elemen array pada indeks j (indeks perulangan untuk pencarian) lebih kecil dari elemen array pada indeks yang ditetapkan sebagai indeks minimum sementara. Jika kondisi terpenuhi / benar maka indeks pada elemen pertama yang digunakan sebagai indeks minimum sementara akan diganti nilainya oleh indeks j (indeks perulangan untuk pencarian)

3) Insertion sort memiliki kondisi pada inner loop (loop untuk swap elemen) dimana perulangan untuk inner loop (swap elemen) akan dilakukan jika indeks j dan nilai elemen pada indeks j (indeks untuk pencarian elemen pada array) memiliki nilai

yang lebih besar daripada nilai pada variabel temp (nilai elemen pada indeks awal / outer loop yang disimpan sebagai nilai sementara).

4) Tujuan dari perintah **data[j + 1] = data[j];** yaitu menduplikasi elemen pada indeks sisi kiri (elemen pada indeks j) ke sebelah kanannya (indeks j+1) sebelum bertukar tempat.

1.2 Percobaan 2: Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

1.2.1 Kode Program

Membuat class baru bernama Mahasiswa11, kemudian menambahkan konstruktor berparameter, atribut, dan method menampilkan informasi mahasiswa.

```
package Jobsheet6;

public class Mahasiswa11 {
    String nim,nama, kelas;
    double ipk;

    Mahasiswa11(){
    }

    Mahasiswa11(String nim, String nama, String kelas, double ipk){
        this.nim = nim;
        this.nama = nama;
        this.ipk = ipk;
        this.kelas = kelas;
    }

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

Kemudian membuat class baru bernama MahasiswaBerprestasi11 dengan atribut array of object dan idx. Kemudian menambahkan method tambah untuk menambahkan data mahasiswa ke array of object method tampil untuk menampilkan daftar mahasiswa berprestasi dan method bubbleSort untuk mengurutkan data mahasiswa prestasi berdasarkan IPK secara descending.

```
package Jobsheet6;

public class MahasiswaBerprestasi11 {
    Mahasiswa11[] listMhs = new Mahasiswa11[5];
    int idx;

    void tambah(Mahasiswa11 mhs){
        if (idx<listMhs.length) {
            listMhs[idx] = mhs;
            idx++;
        }else System.out.println("Data sudah penuh");
    }

    void tampil(){
        for (Mahasiswa11 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) {
                    Mahasiswa11 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}
```

Selanjutnya membuat class MahasiswaDemo11, kemudian membuat array of object dari class MahasiswaBerprestasi11 dan membuat 5 objek dari class Mahasiswa11 kemudian menambahkan semua objek mahasiswa tersebut ke objek dengan memanggil fungsi tambah pada objek MahasiswaBerprestasi. Dan, memanggil method tampil untuk melihat semua data yang telah dimasukan pada MahasiswaBerprestasi, lalu mengurutkan data tersebut dengan menggunakan method bubbleSort dan menampilkan kembali method tampil.

```
package Jobsheet6;

public class MahasiswaDemo11 {
    public static void main(String[] args) {
        MahasiswaBerprestasi11 list = new MahasiswaBerprestasi11();
        Mahasiswa11 m1 = new Mahasiswa11("123", "Zidan", "2A", 3.2);
        Mahasiswa11 m2 = new Mahasiswa11("124", "Ayu", "2A", 3.5);
        Mahasiswa11 m3 = new Mahasiswa11("125", "Sofi", "2A", 3.1);
        Mahasiswa11 m4 = new Mahasiswa11("126", "Sita", "2A", 3.9);
        Mahasiswa11 m5 = new Mahasiswa11("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();
    }
}
```

1.2.2 Hasil Run 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 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
------------------------------
```

1.2.3 Pertanyaan

1) Pada perulangan inner dan perulangan outer yang digunakan untuk algoritma Bubble Sort, diketahui bahwa:
   a. Karena kondisi pada perulangan i (outer loop) menggunakan i < listMhs.length-1 bertujuan agar pada perulangan terakhir hanya membandingkan dua elemen pertama untuk mengecek bahwa seluruh daftar telah diurutkan.
   b. Karena kondisi pada perulangan j (inner loop) menggunakan j < listMhs.length-i bertujuan untuk mengurangi perulangan yang dilakukan agar tidak perlu mengecek kembali elemen yang telah diurutkan.
   c. Jika data pada listMhs adalah 50, maka perulangan i (outer loop) akan dilakukan sebanyak 50-1 sehingga perulangan i akan berjalan sebanyak 49 kali.
2) Memodifikasi program agar data mahasiswa bersifat dinamis (input dari keyborad)

```
package Jobsheet6;

import java.util.Scanner;

public class MahasiswaDemo11 {
    public static void main(String[] args) {
        MahasiswaBerprestasi11 list = new MahasiswaBerprestasi11();
        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < list.listMhs.length; 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();
            sc.nextLine();
            System.out.println("--------------------------------");
            Mahasiswa11 mhs = new Mahasiswa11(nim, nama, kelas, ipk);
            list.tambah(mhs);
        }

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

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

Output kode program

```
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   : 3.3
--------------------------------
```

```
Data mahasiswa sebelum sorting:
Nama: Ali
NIM: 123
Kelas: 2B
IPK: 3.9
--------------------------------
Nama: Ila
NIM: 124
Kelas: 2B
IPK: 3.1
--------------------------------
Nama: Agus
NIM: 125
Kelas: 2B
IPK: 3.6
--------------------------------
Nama: Tika
NIM: 126
Kelas: 2B
IPK: 3.3
--------------------------------
Nama: Udin
NIM: 127
Kelas: 2B
IPK: 3.3
--------------------------------
```

```
Data Mahasiswa setelah sorting berdasarkan IPK (DESC):
Nama: Ali
NIM: 123
Kelas: 2B
IPK: 3.9
--------------------------------
Nama: Agus
NIM: 125
Kelas: 2B
IPK: 3.6
--------------------------------
Nama: Tika
NIM: 126
Kelas: 2B
IPK: 3.3
--------------------------------
Nama: Udin
NIM: 127
Kelas: 2B
IPK: 3.3
--------------------------------
Nama: Ila
NIM: 124
Kelas: 2B
IPK: 3.1
--------------------------------
```

1.3 Percobaan 3: Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

1.3.1 Kode Program

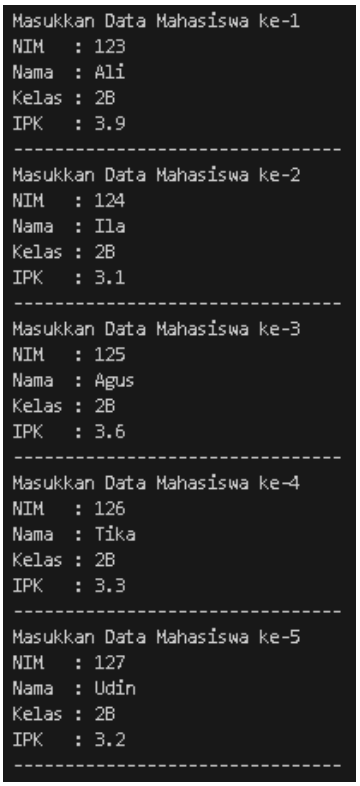
Menambahkan method selectionSort untuk melakukan proses sorting data mahasiswa berprestasi secara ascending menggunakan algoritma Selection Sort pada class MahasiswaBerprestasi11

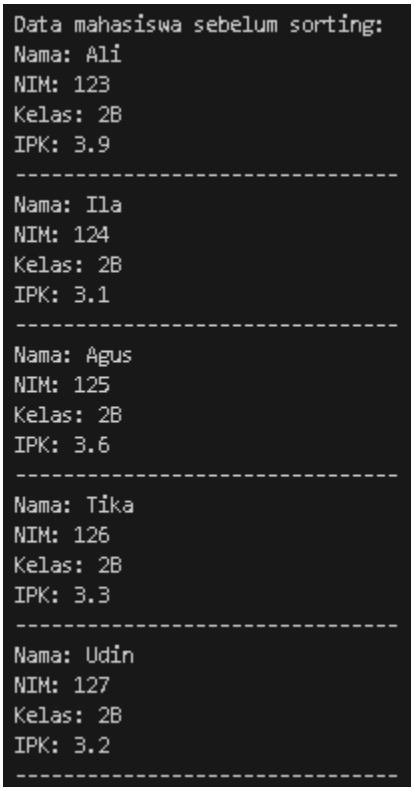
```
void selectionSort(){
    for (int i = 0; i < listMhs.length-1; i++) {
        int idxMin = i;
        for (int j = i+1; j < listMhs.length; j++) {
            if (listMhs[j].ipk<listMhs[idxMin].ipk) {
                idxMin = j;
            }
        }
        Mahasiswa11 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

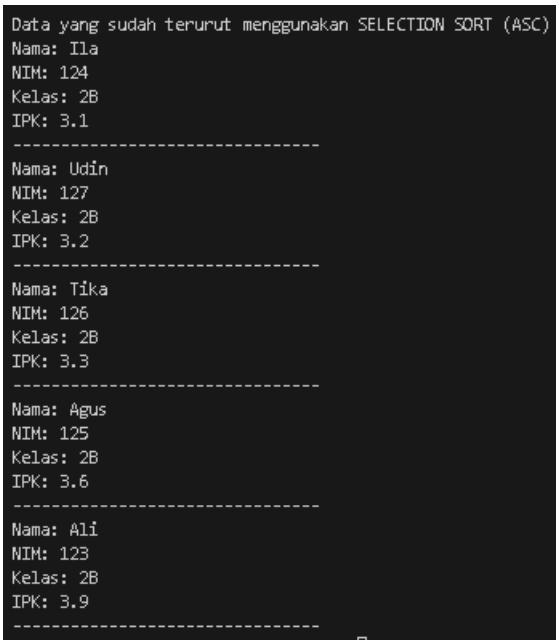
Lalu menambahkan baris program untuk memanggil method selectionSort dan kemudian memanggil method tampil untuk menampilkan data yang telah diurutkan pada class MahasiswaDemo11

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

1.3.2 Hasil Run Program







1.3.3 Pertanyaan

1. Kode tersebut berfungsi untuk melakukan pencarian nilai minimum pada array yang akan disimpan indeks elemennya di variabel idxMin. Kode tersebut memiliki proses

pertama kali menyimpan indeks elemen pertama pada array sebagai nilai minimum sementara, kemudian dilakukan perulangan sebanyak panjang array untuk mengecek apakah elemen array pada indeks j (indeks perulangan untuk pencarian) lebih kecil dari elemen array pada indeks yang ditetapkan sebagai indeks minimum sementara. Jika kondisi terpenuhi / benar maka indeks pada elemen pertama yang digunakan sebagai indeks minimum sementara akan diganti nilainya oleh indeks j (indeks perulangan untuk pencarian)

1.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

1.4.1 Kode Program

Menambahkan method insertionSort untuk melakukan proses sorting data mahasiswa berprestasi secara ascending menggunakan algoritma Insertion Sort pada class MahasiswaBerprestasi11

```
void insertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa11 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Lalu menambahkan baris program untuk memanggil method insertionSort dan kemudian memanggil method tampil untuk menampilkan data yang telah diurutkan pada class MahasiswaDemo11

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

1.4.2 Hasil Run Program

```
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   : 3.5
--------------------------------
```

```
Data mahasiswa sebelum sorting:
Nama: Ayu
NIM: 111
Kelas: 2C
IPK: 3.7
--------------------------------
Nama: Dika
NIM: 222
Kelas: 2C
IPK: 3.0
--------------------------------
Nama: Ila
NIM: 333
Kelas: 2C
IPK: 3.8
--------------------------------
Nama: Susi
NIM: 444
Kelas: 2C
IPK: 3.1
--------------------------------
Nama: Yayuk
NIM: 555
Kelas: 2C
IPK: 3.5
--------------------------------
```

```
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: 3.5
--------------------------------
Nama: Ayu
NIM: 111
Kelas: 2C
IPK: 3.7
--------------------------------
Nama: Ila
NIM: 333
Kelas: 2C
IPK: 3.8
--------------------------------
```

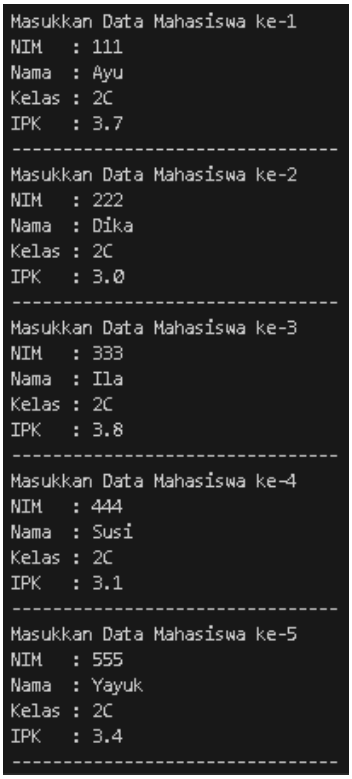
1.4.3 Pertanyaan

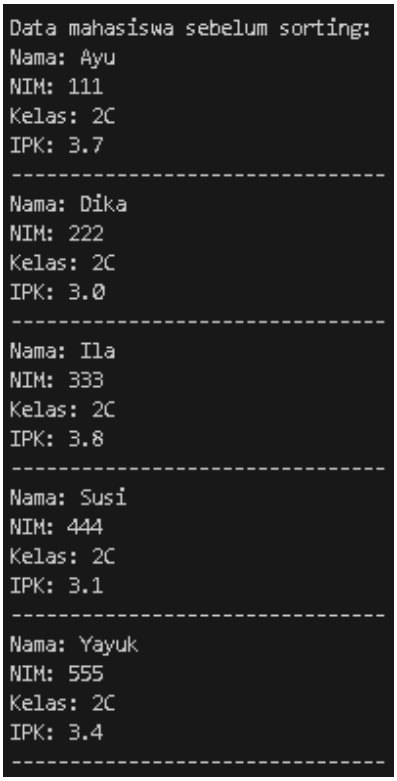
1. Mengubah fungsi pada insertionSort sehingga dapat melakukan sorting secara descending

Kode Program

```
void insertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa11 temp = listMhs[i];
        int j = i;
        while (j>0 && temp.ipk>listMhs[j-1].ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Output Kode Program





```
Data yang sudah terurut menggunakan INSERTION SORT (DESC)
Nama: Ila
NIM: 333
Kelas: 2C
IPK: 3.8
------------------------------
Nama: Ayu
NIM: 111
Kelas: 2C
IPK: 3.7
------------------------------
Nama: Yayuk
NIM: 555
Kelas: 2C
IPK: 3.4
------------------------------
Nama: Susi
NIM: 444
Kelas: 2C
IPK: 3.1
------------------------------
Nama: Dika
NIM: 222
Kelas: 2C
IPK: 3.0
------------------------------
```

1.5 Tugas: Membuat Program untuk menambahkan, Menampilkan dan Mengurutkan data dosen

Kode Program

Class Dosen11

```
package Jobsheet6;

public class Dosen11 {
    String kode, nama;
    Boolean jenisKelamin;
    int usia;
    Dosen11(String kd,String name, Boolean jk,int age){
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

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

Class DataDosen11

```
package Jobsheet6;

public class DataDosen11 {
    Dosen11 [] dataDosen = new Dosen11[10];
    int idx;

    void tambah(Dosen11 dsn){
        if (idx<dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        }else System.out.println("Data sudah penuh");
    }
    void tampil(){
        for (Dosen11 dosen : dataDosen) {
            dosen.tampil();
            System.out.println("--------------------------------");
        }
    }

    void SortingASC(){
        for (int i = 0; i < dataDosen.length-1; i++) {
            for (int j = 1; j < dataDosen.length-i; j++) {
                if (dataDosen[j].usia<dataDosen[j-1].usia) {
                    Dosen11 tmp = dataDosen[j];
                    dataDosen[j] = dataDosen[j-1];
                    dataDosen[j-1] = tmp;
                }
            }
        }
    }

    void sortingDSC(){
        for (int i = 1; i < dataDosen.length; i++) {
            Dosen11 temp = dataDosen[i];
            int j = i;
            while (j>0 && temp.usia>dataDosen[j-1].usia) {
                dataDosen[j] = dataDosen[j-1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}
```

Class MainDosen11

```
package Jobsheet6;

import java.util.Scanner;

public class MainDosen11 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen11 list = new DataDosen11();
        for (int i = 0; i < list.dataDosen.length; i++) {
            System.out.println("Masukkan Data Dosen ke-" + (i + 1));
            System.out.print("Kode Dosen            : ");
            String kode = sc.nextLine();
            System.out.print("Nama                 : ");
            String nama = sc.nextLine();
            System.out.print("Jenis Kelamin (L / P) : ");
            String jenisKelamin = sc.nextLine();
            Boolean jk = jenisKelamin.equalsIgnoreCase("L") ? false : true;
            System.out.print("Usia                 : ");
            int usia = sc.nextInt();
            sc.nextLine();
            System.out.println("--------------------------------");
            Dosen11 dosen = new Dosen11(kode, nama, jk, usia);
            list.tambah(dosen);
        }

        System.out.println("Data dosen sebelum sorting: ");
        list.tampil();

        System.out.println("Data Dosen setelah sorting berdasarkan usia
(ASC): ");
        list.SortingASC();
        list.tampil();

        System.out.println("Data Dosen setelah sorting berdasarkan usia
(DSC): ");
        list.sortingDSC();
        list.tampil();
    }
}
```

Hasil Run Kode Program

```
Masukkan Data Dosen ke-1
Kode Dosen            : A11
Nama                  : Muhammad Ali
Jenis Kelamin (L / P) : L
Usia                  : 32
--------------------------------
Masukkan Data Dosen ke-2
Kode Dosen            : A12
Nama                  : Muhammad Nizam
Jenis Kelamin (L / P) : L
Usia                  : 35
--------------------------------
Masukkan Data Dosen ke-3
Kode Dosen            : A13
Nama                  : Sabrina Rahma
Jenis Kelamin (L / P) : P
Usia                  : 31
--------------------------------
Masukkan Data Dosen ke-4
Kode Dosen            : A14
Nama                  : Muhammad Azzam
Jenis Kelamin (L / P) : L
Usia                  : 39
--------------------------------
Masukkan Data Dosen ke-5
Kode Dosen            : A15
Nama                  : Mika Melatika
Jenis Kelamin (L / P) : P
Usia                  : 41
--------------------------------
Masukkan Data Dosen ke-6
Kode Dosen            : A16
Nama                  : Wahyu Nugraha
Jenis Kelamin (L / P) : L
Usia                  : 29
--------------------------------
Masukkan Data Dosen ke-7
Kode Dosen            : A17
Nama                  : Sita Rahayu
Jenis Kelamin (L / P) : P
Usia                  : 30
--------------------------------
```

```
Masukkan Data Dosen ke-8
Kode Dosen            : 18
Nama                  : Muhammad Reza
Jenis Kelamin (L / P) : L
Usia                  : 40
--------------------------------
Masukkan Data Dosen ke-9
Kode Dosen            : A19
Nama                  : Nur Azizah
Jenis Kelamin (L / P) : P
Usia                  : 43
--------------------------------
Masukkan Data Dosen ke-10
Kode Dosen            : A20
Nama                  : Adam
Jenis Kelamin (L / P) : L
Usia                  : 33
--------------------------------
```

```
Data dosen sebelum sorting:
Kode Dosen    :A11
Nama          :Muhammad Ali
Jenis Kelamin :Laki-laki
Usia          :32
--------------------------------
Kode Dosen    :A12
Nama          :Muhammad Nizam
Jenis Kelamin :Laki-laki
Usia          :35
--------------------------------
Kode Dosen    :A13
Nama          :Sabrina Rahma
Jenis Kelamin :Perempuan
Usia          :31
--------------------------------
Kode Dosen    :A14
Nama          :Muhammad Azzam
Jenis Kelamin :Laki-laki
Usia          :39
--------------------------------
Kode Dosen    :A15
Nama          :Mika Melatika
Jenis Kelamin :Perempuan
Usia          :41
--------------------------------
Kode Dosen    :A16
Nama          :Wahyu Nugraha
Jenis Kelamin :Laki-laki
Usia          :29
--------------------------------
Kode Dosen    :A17
Nama          :Sita Rahayu
Jenis Kelamin :Perempuan
Usia          :30
--------------------------------
Kode Dosen    :A18
Nama          :Muhammad Reza
Jenis Kelamin :Laki-laki
Usia          :40
--------------------------------
```

```
Data Dosen setelah sorting berdasarkan usia (ASC):
Kode Dosen    :A16
Nama          :Wahyu Nugraha
Jenis Kelamin :Laki-laki
Usia          :29
--------------------------------
Kode Dosen    :A17
Nama          :Sita Rahayu
Jenis Kelamin :Perempuan
Usia          :30
--------------------------------
Kode Dosen    :A13
Nama          :Sabrina Rahma
Jenis Kelamin :Perempuan
Usia          :31
--------------------------------
Kode Dosen    :A11
Nama          :Muhammad Ali
Jenis Kelamin :Laki-laki
Usia          :32
--------------------------------
Kode Dosen    :A20
Nama          :Adam
Jenis Kelamin :Laki-laki
Usia          :33
--------------------------------
Kode Dosen    :A12
Nama          :Muhammad Nizam
Jenis Kelamin :Laki-laki
Usia          :35
--------------------------------
Kode Dosen    :A14
Nama          :Muhammad Azzam
Jenis Kelamin :Laki-laki
Usia          :39
--------------------------------
Kode Dosen    :A18
Nama          :Muhammad Reza
Jenis Kelamin :Laki-laki
Usia          :40
--------------------------------
```

```
Kode Dosen    :A19
Nama          :Nur Azizah
Jenis Kelamin :Perempuan
Usia          :43
--------------------------------
Kode Dosen    :A20
Nama          :Adam
Jenis Kelamin :Laki-laki
Usia          :33
--------------------------------
```

```
Kode Dosen    :A15
Nama          :Mika Melatika
Jenis Kelamin :Perempuan
Usia          :41
--------------------------------
Kode Dosen    :A19
Nama          :Nur Azizah
Jenis Kelamin :Perempuan
Usia          :43
--------------------------------
```

```
Data Dosen setelah sorting berdasarkan usia (DSC):
Kode Dosen    :A19
Nama          :Nur Azizah
Jenis Kelamin :Perempuan
Usia          :43
--------------------------------
Kode Dosen    :A15
Nama          :Mika Melatika
Jenis Kelamin :Perempuan
Usia          :41
--------------------------------
Kode Dosen    :A18
Nama          :Muhammad Reza
Jenis Kelamin :Laki-laki
Usia          :40
--------------------------------
Kode Dosen    :A14
Nama          :Muhammad Azzam
Jenis Kelamin :Laki-laki
Usia          :39
--------------------------------
Kode Dosen    :A12
Nama          :Muhammad Nizam
Jenis Kelamin :Laki-laki
Usia          :35
--------------------------------
Kode Dosen    :A20
Nama          :Adam
Jenis Kelamin :Laki-laki
Usia          :33
--------------------------------
Kode Dosen    :A11
Nama          :Muhammad Ali
Jenis Kelamin :Laki-laki
Usia          :32
--------------------------------
Kode Dosen    :A13
Nama          :Sabrina Rahma
Jenis Kelamin :Perempuan
Usia          :31
--------------------------------
```

```
Kode Dosen    :A17
Nama          :Sita Rahayu
Jenis Kelamin :Perempuan
Usia          :30
--------------------------------
Kode Dosen    :A16
Nama          :Wahyu Nugraha
Jenis Kelamin :Laki-laki
Usia          :29
--------------------------------
```

Commit dan Push Kode Program

```
PS C:\Code\Java\sem2\Praktikum-ASD> git commit -m "Jobsheet 6 Tugas Data Dosen"
[main a85f972] Jobsheet 6 Tugas Data Dosen
 3 files changed, 100 insertions(+)
 create mode 100644 Jobsheet5/DataDosen11.java
 create mode 100644 Jobsheet5/Dosen11.java
 create mode 100644 Jobsheet5/MainDosen11.java
PS C:\Code\Java\sem2\Praktikum-ASD> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.47 KiB | 215.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/ghazwanz/Praktikum-ASD.git
   4b102ee..a85f972  main -> main
branch 'main' set up to track 'origin/main'.
```