Niky Tania Sari (18/SIB 1B)

JOBSHEET 5 SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

Praktikum 1 Mengimplementasikan Sorting menggunakan object

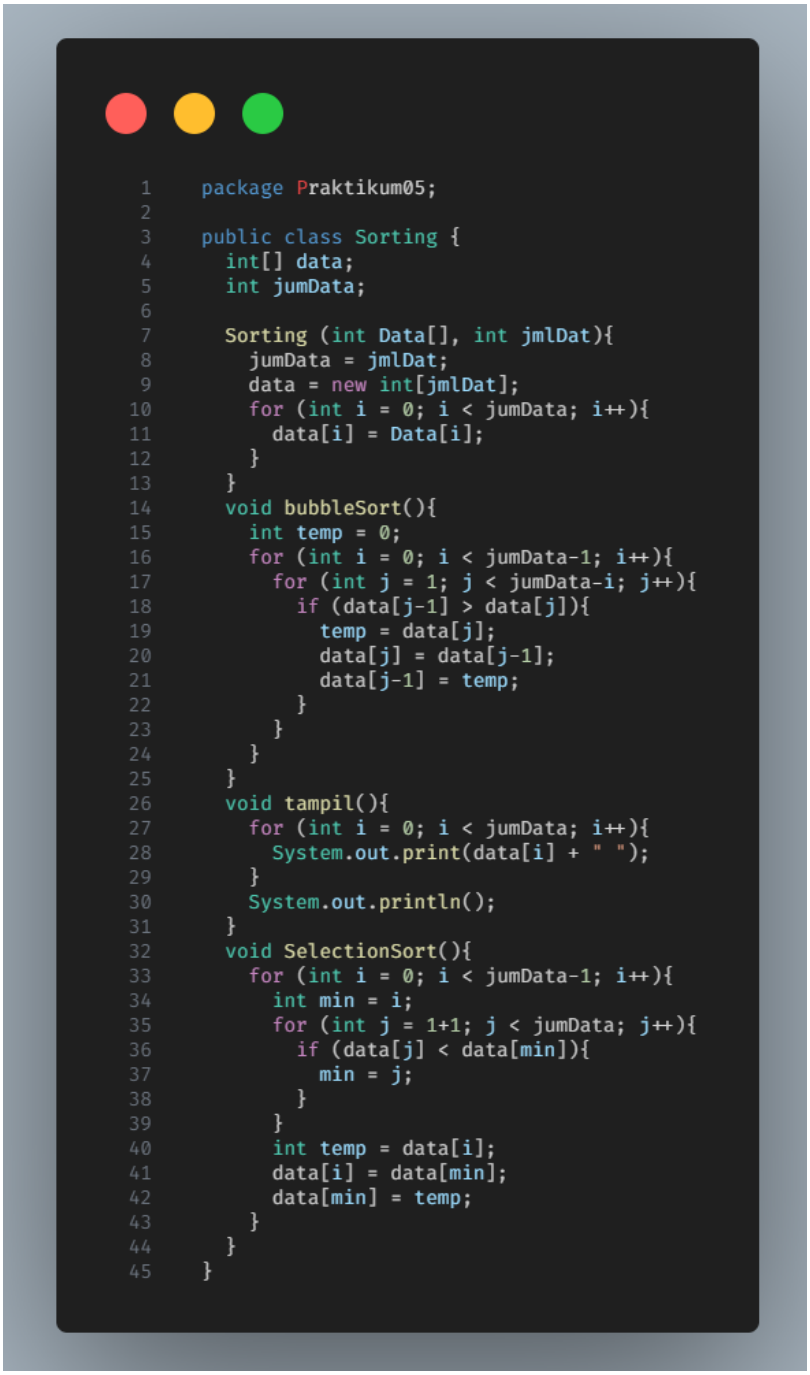
a. SORTING – BUBBLE SORT

```
package Praktikum05;

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

  Sorting (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();
  }
}
```

```
package Praktikum05;

public class SortingMain18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
      int a[] = {20, 10, 2, 7, 12};

    Sorting dataurut1 = new Sorting(a, a.length);

    System.out.println(x:"Data awal 1");
    dataurut1.tampil();
    dataurut1.bubbleSort();
    System.out.println(x:"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
```

b. SORTING – SELECTION SORT



```java
package Praktikum05;

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

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

```java
package Praktikum05;

public class SortingMain18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
      int a[] = {20, 10, 2, 7, 12};
      int b[] = {30, 20, 2, 8, 14};

    Sorting dataurut1 = new Sorting(a, a.length);

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

    Sorting dataurut2 = new Sorting(b, b.length);
    System.out.println(x:"Data awal 2");
    dataurut2.tampil();
    dataurut2.bubbleSort();
    System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
    dataurut2.tampil();

    }
}
```

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

c. SORTING – INSERTION SORT

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

```
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
dataurut1.tampil();

Sorting dataurut2 = new Sorting(b, b.length);
System.out.println(x:"Data awal 2");
dataurut2.tampil();
dataurut2.bubbleSort();
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
dataurut2.tampil();

Sorting dataurut3 = new Sorting(c, c.length);
System.out.println(x:"Data awal 3");
dataurut3.tampil();
dataurut3.bubbleSort();
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
dataurut3.tampil();

}
}
```

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

Pertanyaan 1

1. Jelaskan fungsi kode program berikut

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

Jawaban: Kode tersebut merupakan bagian dari algoritma Bubble Sort yang digunakan untuk mengurutkan array

- Kode ini membandingkan dua elemen yang berdekatan dalam array
- Jika elemen sebelumnya lebih besar dari elemen saat ini, maka dilakukan pertukaran (swap)
- Proses ini berulang hingga elemen terbesar "mengambang" ke posisi akhir array dalam setiap iterasi.

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

Jawaban:

```
for (int i = 0; i < jumData-1; i++){
  int min = i;
  for (int j = 1+1; j < jumData; 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)
```

Jawaban:

- j >= 0 memastikan bahwa indeks tidak keluar dari batas array
- data[j] > temp memastikan bahwa elemen yang lebih besar dari temp akan digeser ke kanan untuk memberi ruang bagi elemen yang sedang disisipkan.

4. Pada Insertion sort, apakah tujuan dari perintah

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

Jawaban:

Setelah elemen yang lebih besar dari temp digeser ke kanan, baris ini memasukkan temp ke dalam posisi yang tepat dalam array yang sudah terurut.

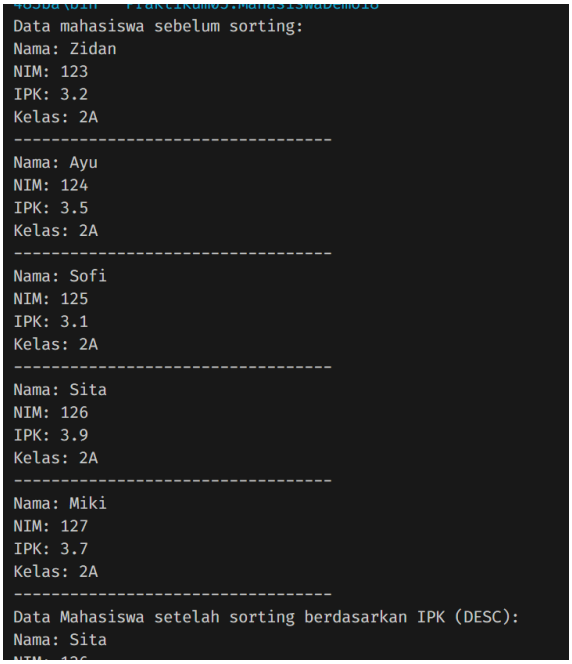
Praktikum 2

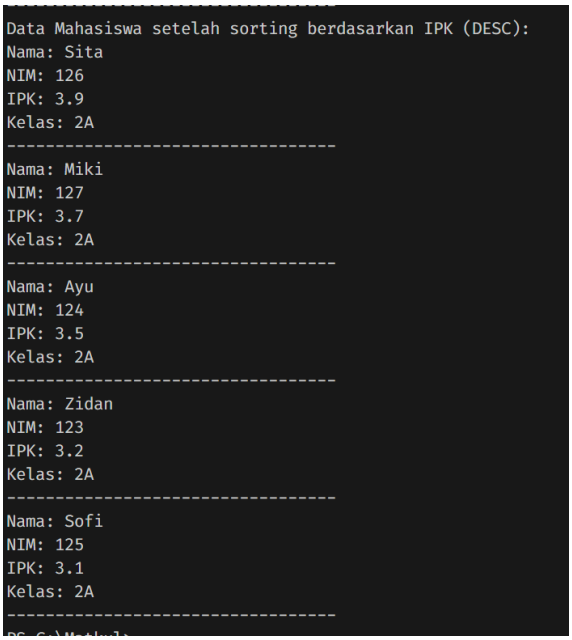
a. (Sorting Menggunakan Array of Object)

```
package Praktikum05;

public class Mahasiswa {
  String nim;
  String nama;
  String kelas;
  double ipk;

  //konstruktor default
  Mahasiswa() {
  }

  //konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama ada yang tidak)
  Mahasiswa(String nm, String name, String kls, double ip){
    nim = nm;
    nama = name;
    ipk = ip;
    kelas = kls;
  }
  void tampilInformasi(){
    System.out.println("Nama: " + nama);
    System.out.println("NIM: " + nim);
    System.out.println("IPK: " + ipk);
    System.out.println("Kelas: " + kelas);
  }
}
```

```java
package Praktikum05;

public class MahasiswaBerprestasi18 {
  Mahasiswa [] listMhs = new Mahasiswa [5];
  int idx;

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

```java
package Praktikum05;

public class MahasiswaDemo18 {
    Run | Debug | Run main | Debug main
  public static void main(String[] args) {
    MahasiswaBerprestasi18 list = new MahasiswaBerprestasi18();
    Mahasiswa m1 = new Mahasiswa(nm:"123",name:"Zidan",kls:"2A",ip:3.2);
    Mahasiswa m2 = new Mahasiswa(nm:"124",name:"Ayu",kls:"2A",ip:3.5);
    Mahasiswa m3 = new Mahasiswa(nm:"125",name:"Sofi",kls:"2A",ip:3.1);
    Mahasiswa m4 = new Mahasiswa(nm:"126",name:"Sita",kls:"2A",ip:3.9);
    Mahasiswa m5 = new Mahasiswa(nm:"127",name:"Miki",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();

  }
}
```





Pertanyaan 2

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 ?
   Jawaban: Perulangan **i** digunakan untuk mengontrol jumlah iterasi proses bubble sort. Karena setiap iterasi memastikan satu elemen terbesar berpindah ke posisi

akhirnya, maka cukup dilakukan sebanyak **listMhs.length - 1** kali. Iterasi terakhir tidak diperlukan karena semua elemen sudah tersusun pada tahap sebelumnya.

b. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?
Jawaban: Perulangan **j** digunakan untuk membandingkan dan menukar elemen dalam array. Pada setiap iterasi **i**, elemen terbesar dari bagian yang belum terurut akan bergeser ke posisi akhirnya. Oleh karena itu, bagian yang sudah terurut tidak perlu dibandingkan lagi, sehingga jumlah iterasi **j** berkurang seiring bertambahnya **i**.

c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?
Jawaban:
- Perulangan i akan berlangsung sebanyak 50 - 1 = 49 kali.
- Tahap bubble sort yang ditempuh juga 49 tahap, karena setiap tahap memastikan satu elemen terbesar sudah berada di posisi akhir yang benar.

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

```java
package Praktikum05;
import java.util.Scanner;

public class MahasiswaDemo18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi18 list = new MahasiswaBerprestasi18();

        System.out.print(s:"Masukkan jumlah mahasiswa: ");
        int jumlahMahasiswa = sc.nextInt();
        sc.nextLine();

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

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

        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();

    }
}
```

```
Masukkan jumlah mahasiswa: 5
Masukkan data mahasiswa ke-1
NIM: 123
Nama: Zidan
IPK: 3.2
Kelas: 2A
Masukkan data mahasiswa ke-2
NIM: 124
Nama: Ayu
IPK: 3.5
Kelas: 2A
Masukkan data mahasiswa ke-3
NIM: 125
Nama: Sofi
IPK: 3.1
Kelas: 2A
Masukkan data mahasiswa ke-4
NIM: 126
Nama: Sita
IPK: 3.9
Kelas: 2A
Masukkan data mahasiswa ke-5
NIM: 127
Nama: Miki
```

```
Nama: Miki
IPK: 3.7
Kelas: 2A
Data mahasiswa sebelum sorting:
Nama: Zidan
NIM: 123
IPK: 3.2
Kelas: 2A
---------------------------------
Nama: Ayu
NIM: 124
IPK: 3.5
Kelas: 2A
---------------------------------
Nama: Sofi
NIM: 125
IPK: 3.1
Kelas: 2A
---------------------------------
Nama: Sita
NIM: 126
IPK: 3.9
Kelas: 2A
---------------------------------
Nama: Miki
```

```
---------------------------------
Nama: Miki
NIM: 127
IPK: 3.7
Kelas: 2A
---------------------------------
Data Mahasiswa setelah sorting berdasarkan IPK (DESC):
Nama: Sita
NIM: 126
IPK: 3.9
Kelas: 2A
---------------------------------
Nama: Miki
NIM: 127
IPK: 3.7
Kelas: 2A
---------------------------------
Nama: Ayu
NIM: 124
IPK: 3.5
Kelas: 2A
---------------------------------
Nama: Zidan
NIM: 123
IPK: 3.2
```

b. 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;
            }
        }
        Mahasiswa tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
      }
    }
  }
```

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

    System.out.println(x:"Data mahasiswa setelah sorting menggunakan SELECTION SORT (ASC):");
    list.selectionSort();
    list.tampil();

  }
}
```

```
Masukkan jumlah mahasiswa: 5
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 sebelum sorting:
Nama: Ali
NIM: 123
IPK: 3.9
Kelas: 2B
---------------------------------
Nama: Ila
NIM: 124
IPK: 3.1
Kelas: 2B
---------------------------------
Nama: agus
NIM: 125
IPK: 3.6
Kelas: 2b
---------------------------------
Nama: tika
NIM: 126
```

```
IPK: 3.3
Kelas: 2b
---------------------------------
Nama: udin
NIM: 127
IPK: 3.2
Kelas: 2b
---------------------------------
Data Mahasiswa setelah sorting berdasarkan IPK (DESC):
Nama: Ali
NIM: 123
IPK: 3.9
Kelas: 2B
---------------------------------
Nama: agus
NIM: 125
IPK: 3.6
Kelas: 2b
---------------------------------
Nama: tika
NIM: 126
IPK: 3.3
Kelas: 2b
---------------------------------
Nama: udin
NIM: 127
IPK: 3.2
Kelas: 2b
---------------------------------
Nama: Ila
```

```
---------------------------------
Nama: Ila
NIM: 124
IPK: 3.1
Kelas: 2B
---------------------------------
Data mahasiswa setelah sorting menggunakan SELECTION SORT (ASC):
Nama: Ila
NIM: 124
IPK: 3.1
Kelas: 2B
---------------------------------
Nama: udin
NIM: 127
IPK: 3.2
Kelas: 2b
---------------------------------
Nama: tika
NIM: 126
IPK: 3.3
Kelas: 2b
---------------------------------
Nama: agus
NIM: 125
IPK: 3.6
Kelas: 2b
---------------------------------
Nama: Ali
NIM: 123
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: berfungsi untuk mencari elemen dengan nilai IPK terkecil dalam array listMhs, mulai dari indeks i hingga akhir array.

1. **int idxMin = i;**:
    - Variabel idxMin digunakan untuk menyimpan indeks dari elemen dengan nilai terkecil (minimum) yang ditemukan selama iterasi.
    - Pada awalnya, idxMin diinisialisasi dengan nilai i, yang merupakan indeks saat ini dalam loop luar. Ini mengasumsikan bahwa elemen pada indeks i adalah yang terkecil.
2. **for (int j = i + 1; j < listMhs.length; j++)**:
    - Loop ini digunakan untuk mencari elemen dengan nilai terkecil (ipk terkecil) di dalam array, dimulai dari indeks i + 1 hingga akhir array (listMhs.length).
    - j adalah indeks yang digunakan untuk iterasi melalui elemen-elemen array.
3. **if (listMhs[j].ipk < listMhs[idxMin].ipk)**:
    - Pada setiap iterasi, program memeriksa apakah nilai ipk dari elemen saat ini (listMhs[j].ipk) lebih kecil daripada nilai ipk dari elemen yang saat ini dianggap terkecil (listMhs[idxMin].ipk).
    - Jika kondisi ini terpenuhi, berarti ditemukan elemen dengan nilai ipk yang lebih kecil.
4. **idxMin = j;**:
    - Jika elemen saat ini (listMhs[j]) memiliki nilai ipk yang lebih kecil, maka idxMin diperbarui dengan indeks j.
    - Dengan demikian, idxMin akan selalu menyimpan indeks dari elemen dengan nilai ipk terkecil yang ditemukan selama iterasi.

Percobaan 3 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

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

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

```
Masukkan jumlah mahasiswa: 5

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
```

```
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)
Nama: dika
NIM: 222
IPK: 3.0
Kelas: 2c
---------------------------------
Nama: susi
NIM: 444
IPK: 3.1
Kelas: 2c
---------------------------------
Nama: yayuk
NIM: 555
IPK: 3.4
Kelas: 2c
---------------------------------
```

```
NIM: 222
IPK: 3.0
Kelas: 2c
---------------------------------
Nama: susi
NIM: 444
IPK: 3.1
Kelas: 2c
---------------------------------
Nama: yayuk
NIM: 555
IPK: 3.4
Kelas: 2c
---------------------------------
Nama: ayu
NIM: 111
IPK: 3.7
Kelas: 2c
---------------------------------
Nama: ila
NIM: 333
IPK: 3.8
Kelas: 2c
---------------------------------
```

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

```
Masukkan jumlah mahasiswa: 5

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
```

```
-----------------------------------
Masukkan data mahasiswa ke-5
NIM: 555
Nama: yayuk
Kelas: 2c
IPK: 3.4
-----------------------------------
Data yang sudah terurut menggunakan INSERTION SORT (ASC)
Nama: ila
NIM: 333
IPK: 3.8
Kelas: 2c
---------------------------------
Nama: ayu
NIM: 111
IPK: 3.7
Kelas: 2c
---------------------------------
Nama: yayuk
NIM: 555
IPK: 3.4
Kelas: 2c
---------------------------------
Nama: susi
NIM: 444
IPK: 3.1
Kelas: 2c
---------------------------------
Nama: dika
```

```
---------------------------------
Nama: dika
NIM: 222
IPK: 3.0
Kelas: 2c
---------------------------------
```

Latihan

```java
package Praktikum05;

public class Dosen {
  String kode;
  String nama;
  Boolean jenisKelamin;
  int usia;


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

  void tampil(){
    System.out.println("Kode: " + kode);
    System.out.println("Nama: " + nama);
    System.out.println("Jenis Kelamin: " + (jenisKelamin ? "L" : "P"));
    System.out.println("Usia: " + usia);
    System.out.println(x:"----------");
  }

}
```

DataDosen.java

```java
package Praktikum05;

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("Data sudah penuh!");
      }
  }

  public void tampil() {
        if (idx == 0) {
            System.out.println("Tidak ada data dosen!");
         return;
        }
        for (int i = 0; i < idx; i++) {
            dataDosen[i].tampil();
        }
    }

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

  public void sortingDSC() { // Selection Sort
      for (int i = 0; i < idx - 1; i++) {
          int idxMax = i;
          for (int j = i + 1; j < idx; j++) {
              if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                  idxMax = j;
              }
          }
          Dosen temp = dataDosen[idxMax];
          dataDosen[idxMax] = dataDosen[i];
          dataDosen[i] = temp;
      }
  }

  public void insertionSort() {
      for (int i = 1; i < idx; i++) {
          Dosen 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;
      }
  }
}
```

DosenMain.java

```java
package Praktikum05;
import java.util.Scanner;

public class DosenMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen data = new DataDosen();

        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 (Bubble Sort)");
            System.out.println("4. Sorting DSC (Selection Sort)");
            System.out.println("5. Sorting DSC (Insertion Sort)");
            System.out.println("6. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine(); // Buang newline agar tidak mengganggu next input

            switch (pilihan) {
                case 1:
                    while (true) { // Perulangan untuk menambah lebih dari satu dosen
                        System.out.println("————————————————————————————————");
                        System.out.print("Masukkan kode dosen: ");
                        String kode = sc.nextLine();
                        System.out.print("Masukkan nama dosen: ");
                        String nama = sc.nextLine();
                        System.out.print("Jenis Kelamin (L/P): ");
                        boolean jk = sc.nextLine().trim().equalsIgnoreCase("L");
                        System.out.print("Masukkan usia dosen: ");
                        int usia = sc.nextInt();
                        sc.nextLine(); // Buang newline

                        Dosen dsn = new Dosen(kode, nama, jk, usia);
                        data.tambah(dsn);

                        System.out.print("Tambah dosen lagi? (Y/N): ");
                        String lanjut = sc.nextLine().trim();
                        if (lanjut.equalsIgnoreCase("N")) {
                            break; // Keluar dari perulangan tambah data
                        }
                    }
                    break;
                case 2:
                    data.tampil();
                    break;
                case 3:
                    data.sortingASC();
                    System.out.println("Data dosen telah diurutkan secara ASCENDING (Usia termuda ke tertua)");
                    data.tampil();
                    break;
                case 4:
                    data.sortingDSC();
                    System.out.println("Data dosen telah diurutkan secara DESCENDING (Selection Sort) Usia tertua ke termuda");
                    data.tampil();
                    break;
                case 5:
                    data.insertionSort();
                    System.out.println("Data dosen telah diurutkan secara DESCENDING (Insertion Sort)");
                    data.tampil();
                    break;
                case 6:
                    System.out.println("Keluar dari program");
                    return; // Hentikan program
                default:
                    System.out.println("Pilihan tidak valid! Coba lagi.");
            }
        }
    }
}
```

```
Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Keluar
Pilih menu: 1
---------------------------------
Masukkan kode dosen: 111
Masukkan nama dosen: Vina
Jenis Kelamin (L/P): P
Masukkan usia dosen: 45
Tambah dosen lagi? (Y/N): y
---------------------------------
Masukkan kode dosen: 222
Masukkan nama dosen: Maura
Jenis Kelamin (L/P): p
Masukkan usia dosen: 50
Tambah dosen lagi? (Y/N): y
---------------------------------
Masukkan kode dosen: 333
Masukkan nama dosen: Reno
Jenis Kelamin (L/P): L
Masukkan usia dosen: 58
```

```
Masukkan usia dosen: 58
Tambah dosen lagi? (Y/N): y
---------------------------------
Masukkan kode dosen: 444
Masukkan nama dosen: dio
Jenis Kelamin (L/P): l
Masukkan usia dosen: 48
Tambah dosen lagi? (Y/N): n

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Keluar
Pilih menu: 2
Kode: 111
Nama: Vina
Jenis Kelamin: P
Usia: 45
----------------------------
Kode: 222
Nama: Maura
Jenis Kelamin: P
Usia: 50
```

```
Jenis Kelamin: P
Usia: 50
----------------------------
Kode: 333
Nama: Reno
Jenis Kelamin: L
Usia: 58
----------------------------
Kode: 444
Nama: dio
Jenis Kelamin: L
Usia: 48
----------------------------

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Keluar
Pilih menu: 3
Data dosen telah diurutkan secara ASCENDING (Usia termuda ke tertua)
Kode: 111
Nama: Vina
Jenis Kelamin: P
```

```
Nama: Vina
Jenis Kelamin: P
Usia: 45
----------------------------
Kode: 444
Nama: dio
Jenis Kelamin: L
Usia: 48
----------------------------
Kode: 222
Nama: Maura
Jenis Kelamin: P
Usia: 50
----------------------------
Kode: 333
Nama: Reno
Jenis Kelamin: L
Usia: 58
----------------------------

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
```

```
----------------------------

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Keluar
Pilih menu: 4
Data dosen telah diurutkan secara DESCENDING (Selection Sort) Usia tertua ke termuda
Kode: 333
Nama: Reno
Jenis Kelamin: L
Usia: 58
----------------------------
Kode: 222
Nama: Maura
Jenis Kelamin: P
Usia: 50
----------------------------
Kode: 444
Nama: dio
Jenis Kelamin: L
Usia: 48
----------------------------
```

```
Jenis Kelamin: L
Usia: 48
----------------------------
Kode: 111
Nama: Vina
Jenis Kelamin: P
Usia: 45
----------------------------

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Keluar
Pilih menu: 5
Data dosen telah diurutkan secara DESCENDING (Insertion Sort)
Kode: 333
Nama: Reno
Jenis Kelamin: L
Usia: 58
----------------------------
Kode: 222
Nama: Maura
Jenis Kelamin: P
```

```
Nama: Maura
Jenis Kelamin: P
Usia: 50
-----------------------------
Kode: 444
Nama: dio
Jenis Kelamin: L
Usia: 48
-----------------------------
Kode: 111
Nama: Vina
Jenis Kelamin: P
Usia: 45
-----------------------------

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Keluar
Pilih menu: 6
Keluar dari program
```