Nama : Deanissa Sherly Sabilla
Kelas : SIB 1B
Absen : 06

JOBSHEET - 5

SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

5.1 Tujuan Praktikum

Setelah melakukan praktikum ini diharapkan mahasiswa mampu:

a. Mahasiswa mampu membuat algoritma searching bubble sort, selection sort dan insertion sort
b. Mahasiswa mampu menerapkan algoritma searching bubble sort, selection sort dan insertion sort pada program

5.2 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort

Waktu : 50 menit

Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa.

Mahasiswa
nama: String thnMasuk: int umur: int ipk: double
Mahasiswa(n: String, t: int, u: int, i: double) tampil(): void

Berdasarkan class diagram di atas, kita akan membuat sebuah class Mahasiswa yang berfunsi untuk membuat objek mahasiswa yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga fungsi tampil() untuk menampilkan semua attribute yang ada.

DaftarMahasiswaBerprestasi
listMhs: Mahasiswa[5] idx: int
tambah(mhs: Mahasiswa): void tampil(): void bubbleSort(): void

Selanjutnya class diagram di atas merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array mahasiswa, misalkan untuk menambahkan objek mahasiswa, menampilkan semua data mahasiswa, dan juga untuk mengurutkan menggunakan Teknik bubble sort berdasarkan nilai IPK mahasiswa.

5.2.1 Langkah-langkah Percobaan

1. Buat project baru dengan nama “bubble-selection-insertion”, kemudian buat package dengan nama “jobsheet6”.
2. Buatlah sebuah class dengan nama Mahasiswa
3. Sesuaikan class Mahasiswa dengan melihat class diagram di atas dengan menambahkan attribute, konstruktor, dan fungsi atau method. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini

```
1   package minggu5;
2
3   public class Mahasiswa {
4       String nama;
5       int thnMasuk, umur;
6       double ipk;
7
8       Mahasiswa(String n, int t, int u, double i){
9           nama = n;
10          thnMasuk = t;
11          umur = u;
12          ipk = i;
13      }
14
15      void tampil(){
16          System.out.println("Nama = "+nama);
17          System.out.println("Tahun Masuk = "+thnMasuk);
18          System.out.println("Umur = "+umur);
19          System.out.println("IPK = "+ipk);
20      }
21  }
```

4. Buat class DaftarMahasiswaBerprestasi seperti di bawah ini!

```
1   package minggu5;
2
3   public class DafraMahasiswaBerprestasi {
4       Mahasiswa listMhs[] = new Mahasiswa[5];
5       int idx;
6
7       //setelah ini tuliskan method tambah()
8
9       //setelah ini tuliskan method tampil()
10
11      //setelah ini tuliskan method bubbleSort()
12  }
```

5. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.

```
7       //setelah ini tuliskan method tambah()
8       void tambah(Mahasiswa m){
9           if(idx<listMhs.length){
10              listMhs[idx] = m;
11              idx++;
12          }else{
13              System.out.println("Data sudah penuh!!");
14          }
15      }
```

6. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.

```
17      //setelah ini tuliskan method tampil()
18      void tampil(){
19          for(Mahasiswa m : listMhs){
20              m.tampil();
21              System.out.println("------------------------");
22          }
23      }
```

7. Tambahkan method bubbleSort() di dalam class tersebut!

```
25      //setelah ini tuliskan method bubbleSort()
26      void bubbleSort(){
27          for(int i=0; i<listMhs.length-1; i++){
28              for(int j=1; j<listMhs.length-i; j++){
29                  if(listMhs[j].ipk > listMhs[j-1].ipk){
30                      //di bawah ini proses swap atau penukaran
31                      Mahasiswa tmp = listMhs[j];
32                      listMhs[j] = listMhs[j-1];
33                      listMhs[j-1] = tmp;
34                  }
35              }
36          }
37      }
```

8. Buat class Main dan didalamnya buat method main() seperti di bawah ini!

```
1   package minggu5;
2   import java.util.Scanner;
3
4   public class Main {
5       public static void main(String[] args) {
6           
7       }
8   }
```

9. **Di dalam method main()**, buatlah sebuah objek DaftarMahasiswaBerprestasi dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek DaftarMahasiswaBerprestasi. Silakan dipanggil fungsi tampil() untuk melihat semua data yang telah dimasukan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.

```
DaftarMahasiswaBerprestasi list = new DaftarMahasiswaBerprestasi();
Mahasiswa m1 = new Mahasiswa("Nusa", 2017, 25, 3);
Mahasiswa m2 = new Mahasiswa("Rara", 2012, 19, 4);
Mahasiswa m3 = new Mahasiswa("Dompu", 2018, 19, 3.5);
Mahasiswa m4 = new Mahasiswa("Abdul", 2017, 23, 2);
Mahasiswa m5 = new Mahasiswa("Ummi", 2019, 21, 3.75);

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 desc berdasarkan ipk");
list.bubbleSort();
list.tampil();
```

5.2.2 Verifikasi Hasil Percobaan

Cocokan hasilnya dengan yang terdapat pada tampilan di bawah ini

```
Data mahasiswa sebelum sorting =
Nama        = Nusa
Tahun Masuk = 2017
Umur        = 25
IPK         = 3.0
-----------------------
Nama        = Rara
Tahun Masuk = 2012
Umur        = 19
IPK         = 4.0
-----------------------
Nama        = Dompu
Tahun Masuk = 2018
Umur        = 19
IPK         = 3.5
-----------------------
Nama        = Abdul
Tahun Masuk = 2017
Umur        = 23
IPK         = 2.0
-----------------------
Nama        = Ummi
Tahun Masuk = 2019
Umur        = 21
IPK         = 3.75
-----------------------
```

```
Data mahasiswa setelah sorting desc berdasarkan ipk
Nama        = Rara
Tahun Masuk = 2012
Umur        = 19
IPK         = 4.0
-----------------------
Nama        = Ummi
Tahun Masuk = 2019
Umur        = 21
IPK         = 3.75
-----------------------
Nama        = Dompu
Tahun Masuk = 2018
Umur        = 19
IPK         = 3.5
-----------------------
Nama        = Nusa
Tahun Masuk = 2017
Umur        = 25
IPK         = 3.0
-----------------------
Nama        = Abdul
Tahun Masuk = 2017
Umur        = 23
IPK         = 2.0
-----------------------
```

- **HASIL INPUT :**

Class Mahasiswa

jobsheet6 > Mahasiswa06.java > Mahasiswa06 > Mahasiswa06(String, int, int, double)

```
4   package jobsheet6;
5   public class Mahasiswa06 {
6   String nama;
7   int thnMasuk, umur;
8   double ipk;
9
10  Mahasiswa06 (String n, int t, int u, double i){
11      nama = n;
12      thnMasuk = t;
13      umur = u;
14      ipk = i;
15  }
16   void tampil (){
17      System.out.println("Nama = "+nama);
18      System.out.println("Tahun Masuk = "+thnMasuk);
19      System.out.println("Umur = "+umur);
20      System.out.println("IPK = "+ipk);
21   }
22
23  }
```

Class Daftar Mahasiswa Berprestasi

jobsheet6 > DaftarMahasiswaBerprestasi06.java > DaftarMahasiswaBerprestasi06 > tampil()

```
3  public class DaftarMahasiswaBerprestasi06 {
4
5      Mahasiswa06 listMhs[] = new Mahasiswa06 [5];
6      int idx;
7      void tambah (Mahasiswa06 m){
8          if (idx<listMhs.length){
9              listMhs[idx] = m;
10             idx++;
11         }else {
12             System.out.println(x:"Data Sudah Penuh!");
13         }
14     }
15     void tampil (){
16         for (Mahasiswa06 m : listMhs){
17             m.tampil();
18             System.out.println(x:"-------------------------");
19         }
20     }
21     void bubbleSort (){
22         for (int i = 0; i < listMhs.length-1; i++) {
23             for (int j = 1; j < listMhs.length-i; j++) {
24                 if (listMhs[j].ipk > listMhs[j-1].ipk){
25                     Mahasiswa06 tmp = listMhs[j];
26                     listMhs[j] = listMhs[j-1];
27                     listMhs[j-1] = tmp;
```

Class Main

jobsheet6 > MahasiswaMain06.java > MahasiswaMain06 > main(String[])

```
3  public class MahasiswaMain06 {
       Run | Debug
4      public static void main(String[] args) {
5          DaftarMahasiswaBerprestasi06 list = new DaftarMahasiswaBerprestasi06();
6          Mahasiswa06 m1 = new Mahasiswa06(n:"Nusa", t:2017, u:25, i:3);
7          Mahasiswa06 m2 = new Mahasiswa06(n:"Rara", t:2012, u:19, i:4);
8          Mahasiswa06 m3 = new Mahasiswa06(n:"Dompu", t:2018, u:19, i:3.5);
9          Mahasiswa06 m4 = new Mahasiswa06(n:"Abdul", t:2017, u:23, i:2);
10         Mahasiswa06 m5 = new Mahasiswa06(n:"Ummi", t:2019, u:21, i:3.75);
11
12         list.tambah (m1);
13         list.tambah (m2);
14         list.tambah (m3);
15         list.tambah (m4);
16         list.tambah (m5);
17
18         System.out.println(x:"Data Mahasiswa sebelum sorting = ");
19         list.tampil();
20
21         System.out.println(x:"Data Mahasiswa setelah s;orting desc berdasarkan ipk");
22         list.bubbleSort();
23         list.tampil();
24     }
25 }
```

➢ **HASIL OUTPUT :**

```
Data Mahasiswa sebelum sorting =
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.0
-------------------------
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.0
-------------------------
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
-------------------------
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
-------------------------
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
-------------------------
```

```
-------------------------
Data Mahasiswa setelah s;orting desc berdasarkan ipk
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.0
-------------------------
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
-------------------------
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
-------------------------
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.0
-------------------------
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
-------------------------
```

5.2.3 Pertanyaan

1. Terdapat di method apakah proses bubble sort?

- Iya, method fungsi `bubbleSort()` merupakan proses bubble sort. bubble sort pada kode program tersebut diimplementasikan dalam dua loop bersarang. Loop pertama digunakan untuk melewati elemen-elemen dalam array, dan loop kedua digunakan untuk membandingkan elemen-elemen berdekatan dan menukar mereka jika diperlukan.

2. Di dalam method bubbleSort(), terdapat baris program seperti di bawah ini:

```
29                  if(listMhs[j].ipk > listMhs[j-1].ipk){
30                      //di bawah ini proses swap atau penukaran
31                      Mahasiswa tmp = listMhs[j];
32                      listMhs[j] = listMhs[j-1];
33                      listMhs[j-1] = tmp;
34                  }
35              }
```

Untuk apakah proses tersebut?

- Proses tersebut adalah langkah pertukaran (swap) dalam bubble sort nilai ipk mahasiswa dari besar ke kecil, langkah ini bertujuan untuk memastikan bahwa elemen yang memiliki nilai yang lebih besar `listMhs[j]` bergerak ke posisi yang lebih tinggi dalam array, sedangkan elemen yang memiliki nilai yang lebih kecil `listMhs[j-1]` bergerak ke posisi yang lebih rendah.

3. Perhatikan perulangan di dalam bubbleSort() di bawah ini:

```
27              for(int i=0; i<listMhs.length-1; i++){
28                  for(int j=1; j<listMhs.length-i; j++){
```

a. Apakah perbedaan antara kegunaan perulangan i dan perulangan j?

- **Perulangan i** digunakan untuk iterasi luar yang menangani satu langkah pengurutan atau "lompatan" yang menempatkan elemen terbesar (atau terkecil) di ujung yang benar dari array
- **Perulangan j** digunakan untuk membandingkan pasangan elemen yang berdekatan dalam setiap iterasi i dan memeriksa dan menukar elemen-elemen yang tidak terurut.

b. Mengapa syarat dari perulangan i adalah i<listMhs.length-1 ?

- Syarat dari perulangan i adalah i < listMhs.length - 1 karena dalam bubble sort, pada setiap iterasi dari perulangan i, dilakukan pengurangan satu pada bagian akhir dari array yang belum terurut.

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

- Syarat dari perulangan j dalam bubble sort adalah j < listMhs.length – i menghindari mempertimbangkan kembali elemen-elemen yang sudah diurutkan dalam setiap iterasi berikutnya, sehingga efisiensi algoritma bubble sort meningkat.

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

- Jika jumlah data dalam listMhs adalah 50, maka perulangan i pada algoritma bubble sort akan berlangsung sebanyak 49 kali. Hal ini karena setiap iterasi luar (iterasi i) akan mengurangi jumlah elemen yang belum terurut di ujung array, dimulai dari 0 hingga 48. Pada iterasi ke-49, hanya ada satu elemen yang perlu dipertimbangkan, sehingga tidak perlu dilakukan iterasi lagi.

5.3 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort

Waktu : 30 menit

Jika pada praktikum yang sebelumnya kita telah mengurutkan data mahasiwa berdasarkan IPK menggunakan Bubble Sort secara descending, pada kali ini kita akan mencoba untuk menambahkan fungsi pengurutan menggunakan Selection Sort.

5.3.1. Langkah-langkah Percobaan.

1. Lihat kembali class DaftarMahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya! Method ini juga akan melakukan proses sorting secara **ascending**, tetapi menggunakan pendekatan selection sort.

```
39      //setelah ini tuliskan method selectionSort()
40      void selectionSort(){
41          for(int i=0; i<listMhs.length-1; i++){
42              int idxMin = i;
43              for(int j=i+1; j<listMhs.length; j++){
44                  if(listMhs[j].ipk < listMhs[idxMin].ipk){
45                      idxMin = j;
46                  }
47              }
48              //swap
49              Mahasiswa tmp = listMhs[idxMin];
50              listMhs[idxMin] = listMhs[i];
51              listMhs[i] = tmp;
52          }
53      }
```

2. Setelah itu, buka kembali class Main, dan di dalam method main() tambahkan baris program untuk memanggil method selectionSort() tersebut!

```
System.out.println("Data mahasiswa setelah sorting asc berdasarkan ipk");
list.selectionSort();
list.tampil();
```

3. Coba jalankan kembali class Main, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

5.3.2. Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

```
Data mahasiswa sebelum sorting =
Nama         = Nusa
Tahun Masuk = 2017
Umur         = 25
IPK          = 3.0
------------------------
Nama         = Rara
Tahun Masuk = 2012
Umur         = 19
IPK          = 4.0
------------------------
Nama         = Dompu
Tahun Masuk = 2018
Umur         = 19
IPK          = 3.5
------------------------
Nama         = Abdul
Tahun Masuk = 2017
Umur         = 23
IPK          = 2.0
------------------------
Nama         = Ummi
Tahun Masuk = 2019
Umur         = 21
IPK          = 3.75
------------------------
```

```
Data mahasiswa setelah sorting asc berdasarkan ipk
Nama         = Abdul
Tahun Masuk = 2017
Umur         = 23
IPK          = 2.0
------------------------
Nama         = Nusa
Tahun Masuk = 2017
Umur         = 25
IPK          = 3.0
------------------------
Nama         = Dompu
Tahun Masuk = 2018
Umur         = 19
IPK          = 3.5
------------------------
Nama         = Ummi
Tahun Masuk = 2019
Umur         = 21
IPK          = 3.75
------------------------
Nama         = Rara
Tahun Masuk = 2012
Umur         = 19
IPK          = 4.0
------------------------
```

➢ **HASIL INPUT :**

Menambahkan **method Selection Sort** pada Class **Daftar Mahasiswa Berprestasi**

```
jobsheet6 > DaftarMahasiswaBerprestasi06.java > DaftarMahasiswaBerprestasi06 > selectionSort()
 3  public class DaftarMahasiswaBerprestasi06 {
32      void selectionSort (){
33          for (int i = 0; i < listMhs.length-1; i++) {
34              int idxMin = i;
35              for (int j = i+1; j < listMhs.length; j++) {
36                  if(listMhs[j].ipk < listMhs[idxMin].ipk){
37                      idxMin = j;
38                  }
39              }
40              Mahasiswa06 tmp = listMhs[idxMin];
41              listMhs[idxMin] = listMhs [i];
42              listMhs[i] = tmp;
43          }
44      }
45  }
46
47
```

➢ **HASIL OUTPUT**

```
Data Mahasiswa sebelum sorting =
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.0
--------------------------
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.0
--------------------------
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
--------------------------
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
--------------------------
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
--------------------------
```

```
Data Mahasiswa setelah sorting asc berdasarkan ipk
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
--------------------------
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.0
--------------------------
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
--------------------------
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
--------------------------
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.0
--------------------------
```

5.3.3. Pertanyaan

Di dalam method selection sort, terdapat baris program seperti di bawah ini:

```
42              int idxMin = i;
43              for(int j=i+1; j<listMhs.length; j++){
44                  if(listMhs[j].ipk < listMhs[idxMin].ipk){
45                      idxMin = j;
46                  }
47              }
```

Untuk apakah proses tersebut, jelaskan!

➢ Proses tersebut adalah langkah selection sort yang digunakan untuk mencari indeks dari elemen terkecil dalam array yang belum diurutkan. (yaitu mengurutkan nilai ipk mahasiswa dari yang terkecil ke terbesar)

5.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

Waktu : 30 menit

Yang terakhir akan diimplementasikan Teknik sorting menggunakan Insertion Sort, dengan mengurutkan IPK mahasiswa secara ascending.

5.4.1 Langkah-langkah Percobaan

1. Lihat kembali class DaftarMahasiswaBerprestasi, dan tambahkan method insertionSort() di dalamnya. Method ini juga akan melakukan proses sorting secara ***ascending***, tetapi menggunakan pendekatan Insertion Sort.

```
68      void insertionSort() {
69          for (int i = 1; i < listMhs.length; i++) {
70              Mahasiswa temp = listMhs[i];
71              int j = i;
72              while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
73                  listMhs[j] = listMhs[j - 1];
74                  j--;
75              }
76              listMhs[j] = temp;
77          }
78      }
```

2. Setelah itu, buka kembali class Main, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() tersebut!

```
System.out.println("Data mahasiswa setelah sorting asc berdasarkan ipk");
list.insertionSort();
list.tampil();
```

3. Coba jalankan kembali class Main, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

- Iya, hasil dari method insertion sort ascending yaitu urut berdasarkan nilai IPK dari terkecil ke terbesar

5.4.2 Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

```
Data mahasiswa sebelum sorting =
Nama        = Nusa
Tahun Masuk = 2017
Umur        = 25
IPK         = 3.0
----------------------
Nama        = Rara
Tahun Masuk = 2012
Umur        = 19
IPK         = 4.0
----------------------
Nama        = Dompu
Tahun Masuk = 2018
Umur        = 19
IPK         = 3.5
----------------------
Nama        = Abdul
Tahun Masuk = 2017
Umur        = 23
IPK         = 2.0
----------------------
Nama        = Ummi
Tahun Masuk = 2019
Umur        = 21
IPK         = 3.75
----------------------
```

```
Data mahasiswa setelah sorting asc berdasarkan ipk
Nama        = Abdul
Tahun Masuk = 2017
Umur        = 23
IPK         = 2.0
----------------------
Nama        = Nusa
Tahun Masuk = 2017
Umur        = 25
IPK         = 3.0
----------------------
Nama        = Dompu
Tahun Masuk = 2018
Umur        = 19
IPK         = 3.5
----------------------
Nama        = Ummi
Tahun Masuk = 2019
Umur        = 21
IPK         = 3.75
----------------------
Nama        = Rara
Tahun Masuk = 2012
Umur        = 19
IPK         = 4.0
----------------------
```

➢ **HASIL INPUT :**

```
    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){
                listMhs[j] = listMhs [j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}
```

➢ **HASIL OUTPUT**

```
Data Mahasiswa sebelum sorting =
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.0
-------------------------
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.0
-------------------------
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
-------------------------
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
-------------------------
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
-------------------------
```

```
Data Mahasiswa setelah sorting asc berdasarkan ipk
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
-------------------------
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.0
-------------------------
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
-------------------------
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
-------------------------
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.0
-------------------------
```

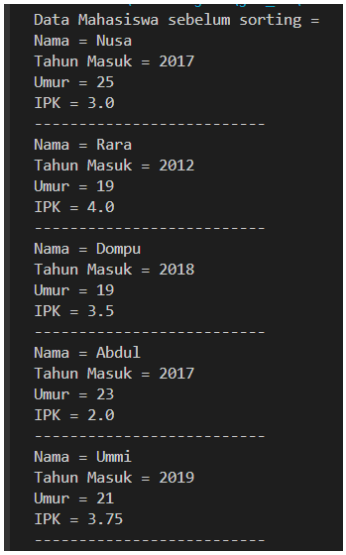
5.4.3 Pertanyaan

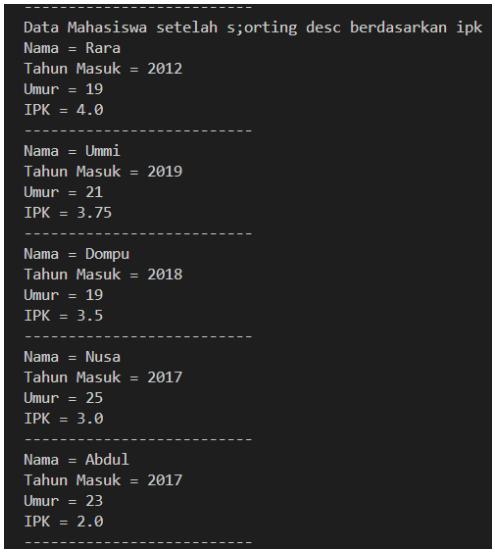
Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.

➢ Tampilan Kode Program **InsertionSort Descending**

```
56      void insertionSort() {
57          for (int i = 1; i < listMhs.length; i++) {
58              Mahasiswa06 temp = listMhs[i];
59              int j = i;
60              while (j > 0 && listMhs[j - 1].ipk < temp.ipk) { // Perubahan disini: mengubah tanda > menjadi <
61                  listMhs[j] = listMhs[j - 1];
62                  j--;
63              }
64              listMhs[j] = temp;
65          }
66      }
67  }
68
```

➢ **Maka Hasil Output**





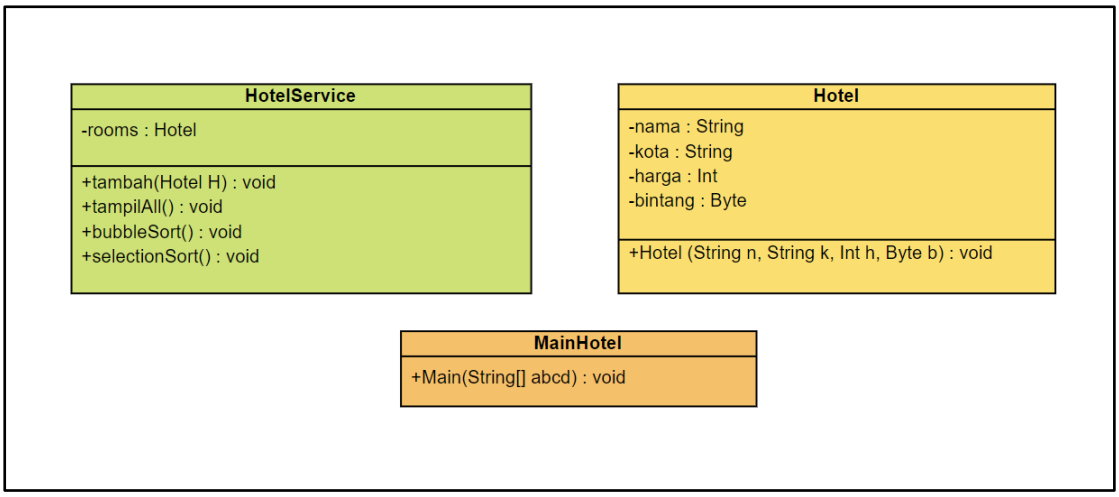
5.5 Latihan Praktikum

Waktu : 90 Menit

Sebuah platform travel yang menyediakan layanan pemesanan kebutuhan travelling sedang mengembangkan backend untuk sistem pemesanan/reservasi akomodasi (penginapan), salah satu fiturnya adalah menampilkan daftar penginapan yang tersedia berdasarkan pilihan filter yang diinginkan user. Daftar penginapan ini harus dapat disorting berdasarkan

1. Harga dimulai dari harga termurah ke harga tertinggi.
2. Rating bintang penginapan dari bintang tertinggi (5) ke terendah (1)

Buatlah proses sorting data untuk kedua filter tersebut dengan menggunakan algoritma **bubble sort** dan **selection sort**.



➢ **HASIL INPUT :**

Class Hotel

```
06.java | DaftarMahasiswaBerprestasi06.java | MahasiswaMain06.java | HotelMain06.java
latihan06 > Hotel06.java > Hotel06 > harga
1  package latihan06;
2
3  public class Hotel06 {
4      String nama, kota;
5      int harga;
6      byte bintang;
7
8      Hotel06 (String n, String k, int h, byte b){
9          nama = n;
10         kota = k;
11         harga = h;
12         bintang = b;
13     }
14     void tampil (){
15         System.out.println("Nama = "+nama);
16         System.out.println("Kota = "+kota);
17         System.out.println("Harga = "+harga);
18         System.out.println("Bintang = "+bintang);
19     }
20 }
```

Class HotelService

latihan06 > HotelService06.java > HotelService06

```java
package latihan06;

public class HotelService06 {
    Hotel06 listHotel[] = new Hotel06 [5];
    int idx;
    //Method Tambah
    void tambah (Hotel06 h){
        if (idx<listHotel.length){
            listHotel[idx] = h;
            idx++;
        }else {
            System.out.println(x:"Data Sudah Penuh!");
        }
    }

    //Method Tampil
    void tampil (){
        for (Hotel06 h : listHotel){
            h.tampil();
            System.out.println(x:"--------------------------");
        }
    }
```

latihan06 > HotelService06.java > HotelService06

```java
public class HotelService06 {
    //Bubble Sort untuk urutan Bintang (Tertinggi ke Terendah)
    void bubbleSort (){
        for (int i = 0; i < listHotel.length-1; i++) {
            for (int j = 1; j < listHotel.length-i; j++) {
                if (listHotel[j].bintang > listHotel[j-1].bintang){
                    Hotel06 tmp = listHotel[j];
                    listHotel[j] = listHotel[j-1];
                    listHotel[j-1] = tmp;
                }
            }
        }
    }
    //Selection Sort untuk urutan Harga (Termurah ke Tertinggi)
    void selectionSort (){
        for (int i = 0; i < listHotel.length-1; i++) {
            int idxMin = i;
            for (int j = i+1; j < listHotel.length; j++) {
                if(listHotel[j].harga < listHotel[idxMin].harga){
                    idxMin = j;
                }
            }
            Hotel06 tmp = listHotel[idxMin];
            listHotel[idxMin] = listHotel [i];
            listHotel[i] = tmp;
        }
```

Class Main

```
latihan06 > HotelMain06.java > HotelMain06 > main(String[])
3   public class HotelMain06 {
        Run | Debug
4       public static void main(String[] args) {
5           HotelService06 list = new HotelService06();
6           Hotel06 h1 = new Hotel06(n:"Nusa", k:"Malang", h:2000, (byte)5);
7           Hotel06 h2 = new Hotel06(n:"Rara", k:"Surabaya", h:2500, (byte)2);
8           Hotel06 h3 = new Hotel06(n:"Agung", k:"Jakarta", h:500, (byte)4);
9           Hotel06 h4 = new Hotel06(n:"Nunung", k:"Bogor", h:1000, (byte)1);
10          Hotel06 h5 = new Hotel06(n:"Yaya", k:"Bandung", h:1500, (byte)3);
11
12          list.tambah (h1);
13          list.tambah (h2);
14          list.tambah (h3);
15          list.tambah (h4);
16          list.tambah (h5);
17
18          System.out.println(x:"Data Hotel sebelum sorting = ");
19          list.tampil();
20
21          System.out.println(x:"Data Hotel setelah sorting DESC berdasarkan Bintang");
22          list.bubbleSort();
23          list.tampil();
24
25          System.out.println(x:"Data Hotel setelah sorting ASC berdasarkan Harga");
26          list.selectionSort();
27          list.tampil();
```

HASIL OUTPUT :

```
PS C:\Users\TOSHIBA\bubble-selection
ava.exe' '-XX:+ShowCodeDetailsInExce
b52ebeba06\redhat.java\jdt_ws\bubble
Data Hotel sebelum sorting =
Nama = Nusa
Kota = Malang
Harga = 2000
Bintang = 5
--------------------------
Nama = Rara
Kota = Surabaya
Harga = 2500
Bintang = 2
--------------------------
Nama = Agung
Kota = Jakarta
Harga = 500
Bintang = 4
--------------------------
Nama = Nunung
Kota = Bogor
Harga = 1000
Bintang = 1
--------------------------
Nama = Yaya
Kota = Bandung
Harga = 1500
Bintang = 3
--------------------------
```

```
--------------------------
Data Hotel setelah sorting DESC berdasarkan Bintang
Nama = Nusa
Kota = Malang
Harga = 2000
Bintang = 5
--------------------------
Nama = Agung
Kota = Jakarta
Harga = 500
Bintang = 4
--------------------------
Nama = Yaya
Kota = Bandung
Harga = 1500
Bintang = 3
--------------------------
Nama = Rara
Kota = Surabaya
Harga = 2500
Bintang = 2
--------------------------
Nama = Nunung
Kota = Bogor
Harga = 1000
Bintang = 1
--------------------------
```

```
--------------------------
Data Hotel setelah sorting ASC berdasarkan Harga
Nama = Agung
Kota = Jakarta
Harga = 500
Bintang = 4
--------------------------
Nama = Nunung
Kota = Bogor
Harga = 1000
Bintang = 1
--------------------------
Nama = Yaya
Kota = Bandung
Harga = 1500
Bintang = 3
--------------------------
Nama = Nusa
Kota = Malang
Harga = 2000
Bintang = 5
--------------------------
Nama = Rara
Kota = Surabaya
Harga = 2500
Bintang = 2
--------------------------
```