

Jobsheet 5

Satria Wiguna/Ti1d/Absen 26

Percobaan 1:

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

Click here to ask Blackbox to help you code faster

```
1 public class DaftarMahasiswaBerprestasi {
2     Mahasiswa listMhs [] = new Mahasiswa[5];
3     int idx;
4
5     void tambah(Mahasiswa m){
6         if(idx<listMhs.length){
7             listMhs[idx] = m;
8             idx++;
9         }else{
10             System.out.println(x:"Data Sudah Penuh");
11         }
12     }
13
14     void tampil() {
15         for (Mahasiswa m : listMhs){
16             m.tampil();
17             System.out.println(x:"-----");
18         }
19     }
20
21
22     void bubbleSort() {
23         for(int i = 0; i < listMhs.length-1; i++) {
24             for(int j = 1; j < listMhs.length-i; j++) {
25                 if(listMhs[j].ipk > listMhs[j-1].ipk) {
26                     Mahasiswa tmp = listMhs[j];
27                     listMhs[j] = listMhs[j-1];
28                     listMhs[j-1] = tmp;
29                 }
30             }
31         }
32     }
33 }
34
```

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

Pertanyaan :

1. Terdapat di method apakah proses bubble sort?
pada kelas DaftarMahasiswaBerprestasi. Pada method tersebut, dilakukan pengurutan (sorting) terhadap array listMhs yang berisi objek Mahasiswa berdasarkan nilai IPK
2. Di dalam method bubbleSort(), terdapat baris program seperti di bawah ini:
Untuk apakah proses tersebut?

```
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     }
```

Proses diatas dilakukan untuk mensorting dari ipk terbesar ke terkecil

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 mengatur iterasi luar, yaitu iterasi yang mengontrol proses keseluruhan bubble sort. Setiap iterasi luar mewakili satu tahap dalam algoritma bubble sort.

Perulangan j digunakan untuk mengatur iterasi dalam, yaitu iterasi yang membandingkan dan menukar elemen-elemen dalam array untuk mengurutkannya.

- b. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?

karena pada setiap iterasi luar, kita akan membandingkan dan menukar elemen terakhir dengan elemen sebelumnya

- c. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?

karena pada setiap iterasi luar, kita ingin mengurangi jumlah iterasi dalam yang diperlukan

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

akan berlangsung sebanyak 49 kali. Ini karena syarat perulangan i adalah $i < \text{listMhs.length} - 1$, yang artinya iterasi luar hanya akan berlangsung selama $\text{listMhs.length} - 1$

Percobaan 2 :

```
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 asc berdasarkan ipk : ");
list.selectionsort();
list.tampil();
}
```

Pernyataan :

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!

untuk mencari indeks dari elemen dengan nilai IPK terkecil dalam rentang tertentu dalam array listMhs. Proses ini dilakukan dengan membandingkan nilai IPK dari setiap elemen dalam rentang tersebut dan memperbarui idxMin dengan IPK terkecil.

Percobaan 3:

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

```
System.out.println(x:"Data mahasiswa setelah sorting asc berdasarkan ipk : ");
list.insertionsort();
list.tampil();
}
```

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

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

Hotel class:

```
1  package Hotel;
2  public class hotel26 {
3      String nama,kota;
4      int harga;
5      Byte bintang;
6      hotel26(String n , String k , int h, byte b){
7          this.nama = n;
8          this.kota = k;
9          this.harga = h;
10         this.bintang = b;
11     }
12
13
14     void tampil(){
15         System.out.println("Nama Kota : " + nama);
16         System.out.println("Kota : " + kota);
17         System.out.println("Harga : " + harga);
18         System.out.println("Rating : " + bintang);
19     }
20
21 }
22
```

Hotel service class:

```
1  package Hotel;
2  public class hotelService26 {
3      hotel26 listRoom[] = new hotel26[4];
4      int idx;
5
6      void tambahHotel(hotel26 h){
7          if(idx < listRoom.length){
8              listRoom[idx] = h;
9              idx++;
10         }else{
11             System.out.println(x:"Data sudah Penuh");
12         }
13     }
14
15     void tampil(){
16         for(hotel26 h : listRoom){
17             h.tampil();
18             System.out.println(x:"-----");
19         }
20     }
21
22     void bubbleSortHrgHotel(){
23         for (int i = 0; i < listRoom.length; i++) {
24             for (int j = 1; j < listRoom.length - i; j++) {
25                 if(listRoom[j].harga < listRoom[j - 1].harga) {
26                     hotel26 tmp = listRoom[j];
27                     listRoom[j] = listRoom[j - 1];
28                     listRoom[j - 1] = tmp;
29                 }
30             }
31         }
32     }
33 }
```

```

31     }
32 }
33
34 void bubbleSortRtHotel(){
35     for (int i = 0; i < listRoom.length; i++) {
36         for (int j = 1; j < listRoom.length - i; j++) {
37             if(listRoom[j].bintang < listRoom[j - 1].bintang) {
38                 hotel26 tmp = listRoom[j];
39                 listRoom[j] = listRoom[j - 1];
40                 listRoom[j - 1] = tmp;
41             }
42         }
43     }
44 }
45
46 void selectionSortHrgHotel(){
47     for (int i = 0; i < listRoom.length - 1; i++) {
48         int idxMin = i;
49         for (int j = i + 1; j < listRoom.length; j++) {
50             if(listRoom[j].harga < listRoom[idxMin].harga) {
51                 idxMin = j;
52             }
53         }
54         hotel26 tmp = listRoom[idxMin];
55         listRoom[idxMin] = listRoom[i];
56         listRoom[i] = tmp;
57     }
58 }

```

```

59
60 void selectionSortRtHotel(){
61     for (int i = 0; i < listRoom.length - 1; i++) {
62         int idxMin = i;
63         for (int j = i + 1; j < listRoom.length; j++) {
64             if(listRoom[j].bintang > listRoom[idxMin].bintang) {
65                 idxMin = j;
66             }
67         }
68         hotel26 tmp = listRoom[idxMin];
69         listRoom[idxMin] = listRoom[i];
70         listRoom[i] = tmp;
71     }
72 }
73 }

```


Main hotel :

```
1  package Hotel;
2  public class mainhotel26 {
    Run | Debug
3      public static void main(String[] args) {
4          hotelService26 list = new hotelService26();
5          hotel26 h1 = new hotel26(n:"Semua Bisa", k:"Malang", h:350000, (byte)3);
6          hotel26 h2 = new hotel26(n:"Jambul Merak", k:"Blitar", h:145000, (byte)2);
7          hotel26 h3 = new hotel26(n:"Shaudagar Khaya", k:"Kediri", h:550000, (byte)5);
8          hotel26 h4 = new hotel26(n:"Kejora kita", k:"Madura", h:90000, (byte)1);
9
10         list.tambahHotel(h1);
11         list.tambahHotel(h2);
12         list.tambahHotel(h3);
13         list.tambahHotel(h4);
14
15         System.out.println(x:"Data Harga Hotel sebelum sorting:");
16         list.tampil();
17         System.out.println(x:"=====");
18         System.out.println(x:"Daftar Hotel Berdasarkan Bintang");
19         System.out.println(x:"Metode Bubble Sort");
20         list.bubbleSortRtHotel();
21         list.tampil();
22         System.out.println(x:"=====");
23         System.out.println(x:"Metode Selection Sort");
24         list.selectionSortRtHotel();
25         list.tampil();
26         System.out.println(x:"=====");
27         System.out.println(x:"Daftar Hotel Berdasarkan Harga");
28         System.out.println(x:"Metode Bubble Sort");
29         list.bubbleSortHrgHotel();
30
31         list.tampil();
32         System.out.println(x:"=====");
33         System.out.println(x:"Metode Selection Sort");
34         list.selectionSortHrgHotel();
35         list.tampil();
36         System.out.println(x:"=====");
37     }
```

Data Harga Hotel sebelum sorting:

Nama Kota : Semua Bisa

Kota : Malang

Harga : 350000

Rating : 3

Nama Kota : Jambul Merak

Kota : Blitar

Harga : 145000

Rating : 2

Nama Kota : Shaudagar Khaya

Kota : Kediri

Harga : 550000

Rating : 5

Nama Kota : Kejora kita

Kota : Madura

Harga : 90000

Rating : 1

=====

=====

Daftar Hotel Berdasarkan Bintang

Metode Bubble Sort

Nama Kota : Kejora kita

Kota : Madura

Harga : 90000

Rating : 1

Nama Kota : Jambul Merak

Kota : Blitar

Harga : 145000

Rating : 2

Nama Kota : Semua Bisa

Kota : Malang

Harga : 350000

Rating : 3

Nama Kota : Shaudagar Khaya

Kota : Kediri

Harga : 550000

Rating : 5

```
=====
Metode Selection Sort
Nama Kota : Shaudagar Khaya
Kota : Kediri
Harga : 550000
Rating : 5
-----
```

```
Nama Kota : Semua Bisa
Kota : Malang
Harga : 350000
Rating : 3
-----
```

```
Nama Kota : Jambul Merak
Kota : Blitar
Harga : 145000
Rating : 2
-----
```

```
Nama Kota : Kejora kita
Kota : Madura
Harga : 90000
Rating : 1
-----
```

```
=====
Daftar Hotel Berdasarkan Harga
Metode Bubble Sort
Nama Kota : Kejora kita
Kota : Madura
Harga : 90000
Rating : 1
-----
```

```
Nama Kota : Jambul Merak
Kota : Blitar
Harga : 145000
Rating : 2
-----
```

```
Nama Kota : Semua Bisa
Kota : Malang
Harga : 350000
Rating : 3
-----
```

```
Nama Kota : Shaudagar Khaya
Kota : Kediri
Harga : 550000
Rating : 5
-----
```

```
=====
Metode Selection Sort
Nama Kota : Kejora kita
Kota : Madura
Harga : 90000
Rating : 1
-----
Nama Kota : Jambul Merak
Kota : Blitar
Harga : 145000
Rating : 2
-----
Nama Kota : Semua Bisa
Kota : Malang
Harga : 350000
Rating : 3
-----
Nama Kota : Shaudagar Khaya
Kota : Kediri
Harga : 550000
Rating : 5
-----
```