

Jobsheet 5

Praktikum Algoritma & Struktur Data



Rafi Ody Prasetyo
(2341720180)

D-IV Teknik Informatika
Politeknik Negeri Malang
Semester 2
2024

Percobaan 1

Code:

Mahasiswa21.java

```
1 public class Mahasiswa21 {
2
3     public String nama;
4     public int thnMasuk, umur;
5     public double ipk;
6
7     public Mahasiswa21(String n, int t, int u, double i)
8     {
9         nama = n;
10        thnMasuk = t;
11        umur = u;
12        ipk = i;
13    }
14
15    public void tampil()
16    {
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 }
```

DaftarMahasiswaBeprestasi21.java

```
1 public class DaftarMahasiswaBerprestasi21 {
2
3     public Mahasiswa21 listMhs[] = new Mahasiswa21[5];
4     public int idx;
5
6     public void tambah(Mahasiswa21 m)
7     {
8         if (idx < listMhs.length)
9         {
10            listMhs[idx] = m;
11            idx++;
12        }
13        else
14        {
15            System.out.println(x:"Data sudah penuh!!");
16        }
17    }
18
19    public void tampil()
20    {
21        for (Mahasiswa21 m : listMhs) {
22            m.tampil();
23            System.out.println(x:"-----");
24        }
25    }
26 }
```

```

27     public void bubbleSort()
28     {
29         for (int i = 0; i < listMhs.length-1; i++) {
30             for (int j = 1; j < listMhs.length-i; j++) {
31                 if (listMhs[j].ipk > listMhs[j-1].ipk) {
32                     Mahasiswa21 tmp = listMhs[j];
33                     listMhs[j] = listMhs[j-1];
34                     listMhs[j-1] = tmp;
35                 }
36             }
37         }
38     }
39 }
40

```

MainMahasiswa21.java

```

1  public class MainMahasiswa21 {
2
3      Run | Debug
4      public static void main(String[] args) {
5
6          DaftarMahasiswaBerprestasi21 list = new DaftarMahasiswaBerprestasi21();
7          Mahasiswa21 m1 = new Mahasiswa21(n:"Nusa", t:2017, u:25, i:3);
8          Mahasiswa21 m2 = new Mahasiswa21(n:"Rara", t:2012, u:19, i:4);
9          Mahasiswa21 m3 = new Mahasiswa21(n:"Dompur", t:2018, u:19, i:3.5);
10         Mahasiswa21 m4 = new Mahasiswa21(n:"Abdul", t:2017, u:23, i:2);
11         Mahasiswa21 m5 = new Mahasiswa21(n:"Ummi", t:2019, u:21, i:3.75);
12
13         list.tambah(m1);
14         list.tambah(m2);
15         list.tambah(m3);
16         list.tambah(m4);
17         list.tambah(m5);
18
19         System.out.println(x:"Data Mahasiswa Sebelum Sorting: ");
20         list.tampil();
21
22         System.out.println(x:"Data Mahasiswa Setelah Sorting desc Berdasarkan IPK: ");
23         list.bubbleSort();
24         list.tampil();
25     }
26 }
27

```

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: Dompur
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: Dampu
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

Pertanyaan

1. Terdapat di method apakah proses bubble sort?

Jawab:

Bubble sort terdapat pada method void bubbleSort() yang terdapat pada class DaftarMahasiswaBerprestasi.

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?

Jawab:

Kondisi pada statement if digunakan untuk memeriksa apakah IPK dari mahasiswa pada posisi j lebih besar dari IPK mahasiswa pada posisi j-1. Jika iya, maka artinya elemen pada posisi j perlu ditukar dengan elemen pada posisi j-1, sehingga elemen dengan IPK lebih tinggi akan bergerak ke arah kanan array. Kemudian untuk variabel tmp digunakan untuk menyimpan elemen yang akan dipindahkan (elemen pada posisi j) ke dalam variabel sementara tmp. Selanjutnya listMhs[j] = listMhs[j-1]; digunakan untuk menyalin elemen pada posisi j-1 ke posisi j, karena elemen pada posisi j akan digantikan dengan elemen yang lebih kecil. Dan yang terakhir listMhs[j-1] = tmp; digunakan untuk menempatkan kembali elemen yang disimpan dalam variabel

sementara (tmp) ke posisi j-1, sehingga elemen pada posisi j-1 sekarang berisi elemen yang lebih besar.

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?

Jawab:

Outer loop digunakan untuk melakukan iterasi sejumlah yang diperlukan untuk mengurutkan array. Sedang inner loop digunakan untuk melakukan perbandingan dan penukaran elemen array. Loop akan berjalan dari 1 hingga listMhs.length - 1. Hal ini dilakukan untuk memastikan bahwa iterasi loop dalam tidak akan melebihi batas yang telah diurangi dari panjang array dan iterasi loop luar. Dengan demikian, pada setiap iterasi dari loop luar, satu elemen besar (atau kecil, tergantung pada urutan yang diinginkan) akan diposisikan dengan benar di akhir array.

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

Jawab:

Pada setiap iterasi dari perulangan i, algoritma Bubble Sort memastikan bahwa elemen terbesar (atau terkecil, tergantung pada urutan yang diinginkan) telah berpindah ke posisi yang benar di ujung array. Jika kita terus melanjutkan proses pengurutan setelah elemen terakhir berada di posisi yang benar, itu hanya akan membuang waktu karena tidak akan ada penukaran yang terjadi.

c) Mengapa syarat dari perulangan j adalah j<listMhs.length-1 ?

Jawab:

Agar iterasi tidak membandingkan elemen yang sudah diurutkan.

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

Jawab:

Perulangan i akan melakukan iterasi sebanyak 49 kali, dan melakukan bubble sort sebanyak maksimal 1225 tahap.

Percobaan 2

Code:

DaftarMahasiswaBerprestasi21.java

```
public 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;
            }
        }
        Mahasiswa21 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

Output:

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: Dompus

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

Pertanyaan

1. 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!

Jawab:

Proses tersebut berfungsi untuk mengurutkan nilai dalam array listMhs dari yang terkecil ke terbesar berdasarkan nilai ipk. Variabel idxMin digunakan untuk nilai awal dengan nilai i. Kemudian untuk variabel j digunakan untuk membandingkan nilai dengan nilai variabel idxMin. Jika nilai j lebih kecil dari nilai variabel idxMin maka nilai idxMin akan diganti dengan j.

Percobaan 4

Code:

DaftarMahasiswaBerprestasi21.java

```
public void insertionSort()
{
    for (int i = 0; i < listMhs.length; i++) {
        Mahasiswa21 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Output:

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

Pertanyaan

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

Jawab:

Before:

```
public void insertionSort()
{
    for (int i = 0; i < listMhs.length; i++) {
        Mahasiswa21 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

After:


```

public void insertionSort()
{
    for (int i = 0; i < listMhs.length; i++) {
        Mahasiswa21 temp = listMhs[i];
        int j = i;
        // while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
        while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}

```

Output:

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

Praktikum

Code:

Hotel21.java

```
1 public class Hotel21 {
2
3     public String nama, kota;
4     public int harga;
5     public byte bintang;
6
7     public Hotel21 (String n, String k, int h, byte b)
8     {
9         this.nama = n;
10        this.kota = k;
11        this.harga = h;
12        this.bintang = b;
13    }
14 }
```

HotelService21.java

```
1 public class HotelService21 {
2
3     public Hotel21 rooms[] = new Hotel21[5];
4     public int idx;
5
6     public void tambah(Hotel21 h)
7     {
8         if (idx < rooms.length) {
9             rooms[idx] = h;
10            idx++;
11        }
12        else
13        {
14            System.out.println(x:"Data sudah penuh!!");
15        }
16    }
17
18    public void tampilAll()
19    {
20        for (Hotel21 h : rooms) {
21            System.out.println("Nama: "+ h.nama);
22            System.out.println("Kota: "+ h.kota);
23            System.out.println("Harga: "+ h.harga);
24            System.out.println("Bintang: "+ h.bintang);
25            System.out.println(x:"-----");
26        }
27    }
28 }
```

```

29     public void bubbleSortHrg()
30     {
31         for (int i = 0; i < rooms.length - 1; i++) {
32             for (int j = 1; j < rooms.length - i; j++) {
33                 if (rooms[j].harga < rooms[i].harga) {
34                     Hotel21 temp = rooms[j];
35                     rooms[j] = rooms[j - 1];
36                     rooms[j - 1] = temp;
37                 }
38             }
39         }
40     }
41
42     public void bubbleSortRtg()
43     {
44         for (int i = 0; i < rooms.length - 1; i++) {
45             for (int j = 1; j < rooms.length - i; j++) {
46                 if (rooms[j].bintang > rooms[i].bintang) {
47                     Hotel21 temp = rooms[j];
48                     rooms[j] = rooms[j - 1];
49                     rooms[j - 1] = temp;
50                 }
51             }
52         }
53     }
54

```

```

55     public void selectionSortHrg()
56     {
57         for (int i = 0; i < rooms.length; i++) {
58             int idxMin = i;
59             for (int j = i+1; j < rooms.length; j++) {
60                 if (rooms[j].harga < rooms[idxMin].harga) {
61                     idxMin = j;
62                 }
63             }
64             Hotel21 tmp = rooms[idxMin];
65             rooms[idxMin] = rooms[i];
66             rooms[i] = tmp;
67         }
68     }
69
70     public void selectionSortRtg()
71     {
72         for (int i = 0; i < rooms.length; i++) {
73             int idxMin = i;
74             for (int j = i+1; j < rooms.length; j++) {
75                 if (rooms[j].bintang > rooms[idxMin].bintang) {
76                     idxMin = j;
77                 }
78             }
79             Hotel21 tmp = rooms[idxMin];
80             rooms[idxMin] = rooms[i];
81             rooms[i] = tmp;
82         }
83     }
84
85 }

```

MainHotel21.java

```
1 public record MainHotel21() {
2
3     Run | Debug
4     public static void main(String[] args) {
5
6         HotelService21 list = new HotelService21();
7         Hotel21 h1 = new Hotel21(n:"Oyo", k:"Malang", h:25000, (byte) 3);
8         Hotel21 h2 = new Hotel21(n:"Airi", k:"Surabaya", h:10000, (byte) 4);
9         Hotel21 h3 = new Hotel21(n:"Santika", k:"Jakarta", h:19000, (byte) 5);
10        Hotel21 h4 = new Hotel21(n:"RedDoorz", k:"Jember", h:23000, (byte) 2);
11        Hotel21 h5 = new Hotel21(n:"Mercure", k:"Bogor", h:21000, (byte) 3);
12
13        list.tambah(h1);
14        list.tambah(h2);
15        list.tambah(h3);
16        list.tambah(h4);
17        list.tambah(h5);
18
19        System.out.println(x:"Data Hotel sebelum di sorting: ");
20        list.tampilAll();
21
22        System.out.println(x:"Data harga hotel asc bubble sort: ");
23        list.bubbleSortHrg();
24        list.tampilAll();
25
26        System.out.println(x:"Data rating hotel desc bubble sort: ");
27        list.bubbleSortRtg();
28        list.tampilAll();
29
30        System.out.println(x:"Data harga hotel asc selection sort: ");
31        list.selectionSortHrg();
32        list.tampilAll();
33
34        System.out.println(x:"Data rating hotel desc selection sort: ");
35        list.selectionSortRtg();
36        list.tampilAll();
37
38    }
39 }
```

Output:

Data Hotel sebelum di sorting:

```
Nama: Oyo
Kota: Malang
Harga: 25000
Bintang: 3
-----
Nama: Airi
Kota: Surabaya
Harga: 10000
Bintang: 4
-----
Nama: Santika
Kota: Jakarta
Harga: 19000
Bintang: 5
-----
Nama: RedDoorz
Kota: Jember
Harga: 23000
Bintang: 2
-----
Nama: Mercure
Kota: Bogor
Harga: 21000
Bintang: 3
-----
```

Data harga hotel asc bubble sort:

```
Nama: Airi
Kota: Surabaya
Harga: 10000
Bintang: 4
-----
Nama: Santika
Kota: Jakarta
Harga: 19000
Bintang: 5
-----
Nama: Oyo
Kota: Malang
Harga: 25000
Bintang: 3
-----
Nama: RedDoorz
Kota: Jember
Harga: 23000
Bintang: 2
-----
Nama: Mercure
Kota: Bogor
Harga: 21000
Bintang: 3
-----
```

Data rating hotel desc bubble sort:

Nama: Santika

Kota: Jakarta

Harga: 19000

Bintang: 5

Nama: Airi

Kota: Surabaya

Harga: 10000

Bintang: 4

Nama: Oyo

Kota: Malang

Harga: 25000

Bintang: 3

Nama: RedDoorz

Kota: Jember

Harga: 23000

Bintang: 2

Nama: Mercure

Kota: Bogor

Harga: 21000

Bintang: 3

Data harga hotel asc selection sort:

Nama: Airi

Kota: Surabaya

Harga: 10000

Bintang: 4

Nama: Santika

Kota: Jakarta

Harga: 19000

Bintang: 5

Nama: Mercure

Kota: Bogor

Harga: 21000

Bintang: 3

Nama: RedDoorz

Kota: Jember

Harga: 23000

Bintang: 2

Nama: Oyo

Kota: Malang

Harga: 25000

Bintang: 3

Data rating hotel desc selection sort:

Nama: Santika

Kota: Jakarta

Harga: 19000

Bintang: 5

Nama: Airi

Kota: Surabaya

Harga: 10000

Bintang: 4

Nama: Mercure

Kota: Bogor

Harga: 21000

Bintang: 3

Nama: Oyo

Kota: Malang

Harga: 25000

Bintang: 3

Nama: RedDoorz

Kota: Jember

Harga: 23000

Bintang: 2
