

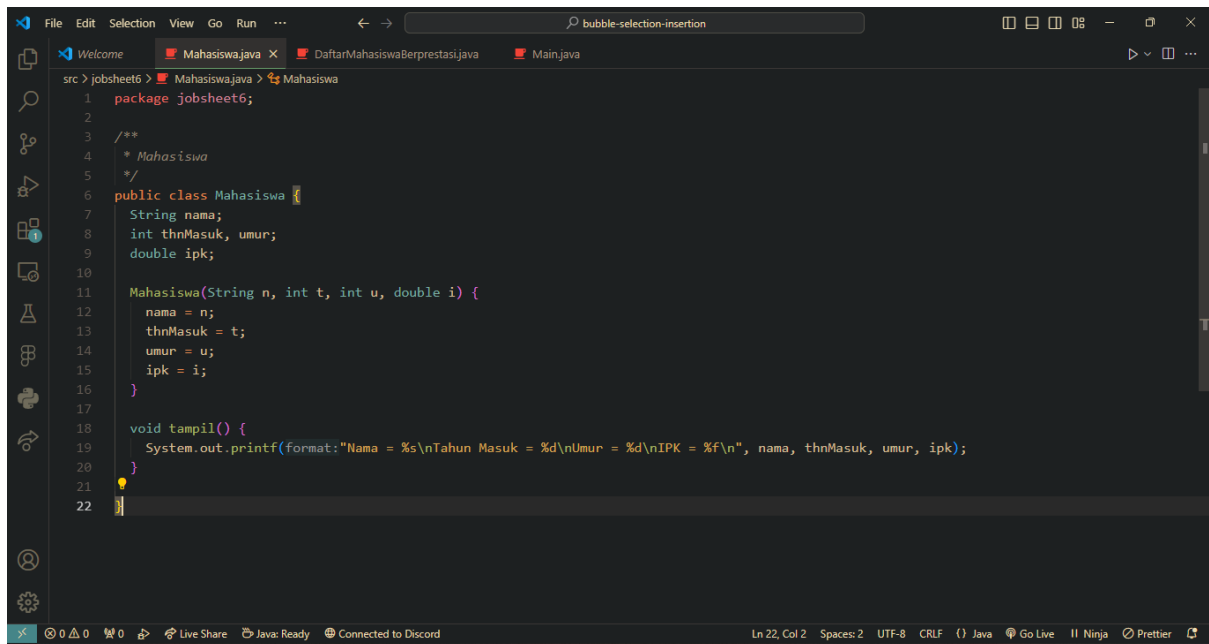
# ALGORITMA STRUKTUR DATA

## Praktikum – Sorting

Lavina 2341760062

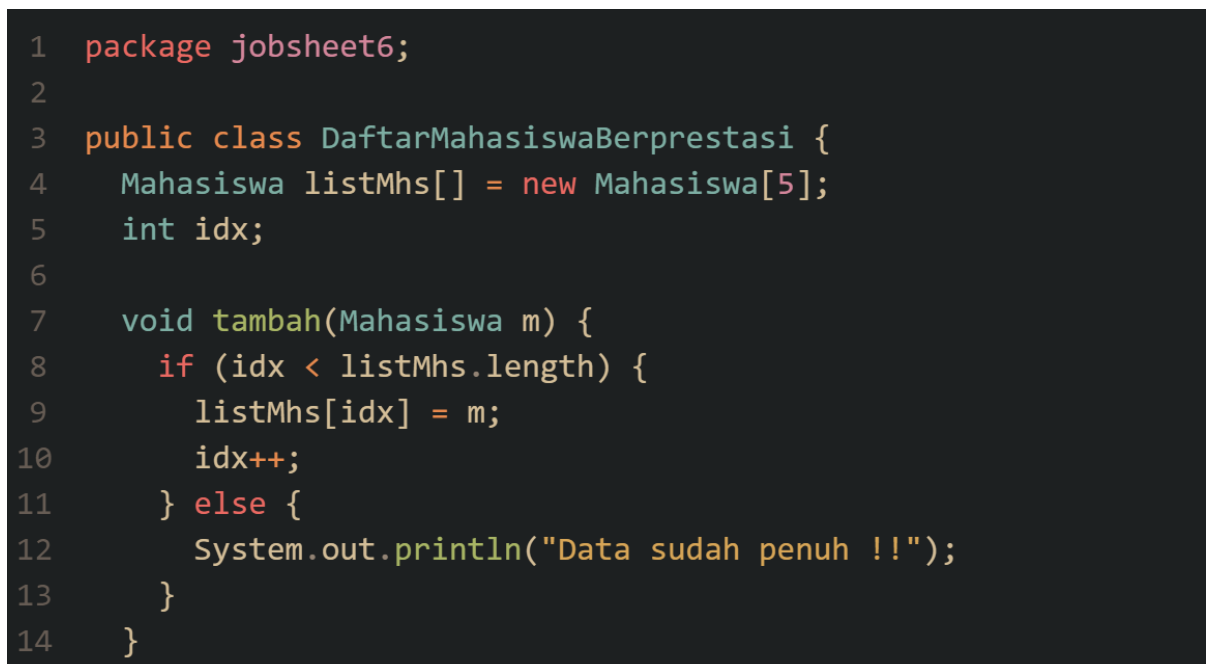
### Praktikum 1: Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort

Mahasiswa.java



```
1 package jobsheet6;
2
3 /**
4  * Mahasiswa
5  */
6 public class Mahasiswa {
7     String nama;
8     int thnMasuk, umur;
9     double ipk;
10
11     Mahasiswa(String n, int t, int u, double i) {
12         nama = n;
13         thnMasuk = t;
14         umur = u;
15         ipk = i;
16     }
17
18     void tampil() {
19         System.out.printf(format:"Nama = %s\nTahun Masuk = %d\nUmur = %d\nIPK = %f\n", nama, thnMasuk, umur, ipk);
20     }
21
22 }
```

DaftarMahasiswaBerprestasi.java



```
1 package jobsheet6;
2
3 public class DaftarMahasiswaBerprestasi {
4     Mahasiswa listMhs[] = new Mahasiswa[5];
5     int idx;
6
7     void tambah(Mahasiswa m) {
8         if (idx < listMhs.length) {
9             listMhs[idx] = m;
10            idx++;
11        } else {
12            System.out.println("Data sudah penuh !!");
13        }
14    }
15 }
```

```

16 void tampil() {
17     for (Mahasiswa m : listMhs) {
18         m.tampil();
19         System.out.println("=====");
20     }
21 }
22
23 void bubbleSort() {
24     for (int i = 0; i < listMhs.length - 1; i++) {
25         for (int j = 1; j < listMhs.length - i; j++) {
26             if (listMhs[j].ipk > listMhs[j - 1].ipk) {
27                 Mahasiswa tmp = listMhs[j];
28                 listMhs[j] = listMhs[j - 1];
29                 listMhs[j - 1] = tmp;
30             }
31         }
32     }
33 }
34 }

```

Output:

```

Data Mahasiswa sebelum sorting =
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.00
=====
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.00
=====
Nama = Dompur
Tahun Masuk = 2018
Umur = 19
IPK = 3.50
=====
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.00
=====
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 = 4.00
=====
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
=====
Nama = Dompur
Tahun Masuk = 2018
Umur = 19
IPK = 3.50
=====
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.00
=====
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.00
=====

```

## Pertanyaan

1. Terdapat di method apakah proses bubble sort?

**Jawab:** Terdapat di method **bubbleSort()**

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 :** Sintaks itu mengecek apakah ipk pada indeks j lebih besar dari ipk indeks j – 1 atau ipk yang ada disebelah kanan, jika iya, maka akan diswap dengan cara mengisi value mahasiswa j dengan value dari mahasiswa j – 1. Lalu mahasiswa j – 1 akan diisi dengan value tmp yang sudah diinisialisasi sebelumnya dengan isi value mahasiswa j.

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 :** Perulangan i untuk mengakses objek mahasiswa yang diinginkan sedangkan perulangan j untuk mengakses atribut dari objek mahasiswa.

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

**Jawab :** Agar tidak mengalami error `indexoutofbound`. Jika panjang arraynya 5 maka `length`-nya harus dikurangi 1 karena indexnya hanya sampai 4.

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

**Jawab :** Agar tidak terjadi error `indexoutofbounds`.

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

**Jawab :** Ada 49 tahap, karena perulangan i akan terus berjalan hingga `length` dari `listMhs` – 1 sehingga  $50 - 1 = 49$ .

## Percobaan 2: Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

DaftarMahasiswaBerprestasi.java

```
1  void selectionSort() {
2      for (int i = 0; i < listMhs.length - 1; i++) {
3          int idxMin = i;
4          for (int j = i + 1; j < listMhs.length; j++) {
5              if (listMhs[j].ipk < listMhs[idxMin].ipk) {
6                  idxMin = j;
7              }
8              Mahasiswa tmp = listMhs[idxMin];
9              listMhs[idxMin] = listMhs[i];
10             listMhs[i] = tmp;
11         }
12     }
13 }
```

Main.java

```
1  System.out.println("Data Mahasiswa setelah sorting asc berdasarkan IPK");
2  list.selectionSort();
3  list.tampil();
```

Output:

```
Data Mahasiswa setelah sorting asc berdasarkan IPK
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.00
=====
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.00
=====
Nama = Dompus
Tahun Masuk = 2018
Umur = 19
IPK = 3.50
=====
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
=====
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.00
=====
```

## 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!

**Jawab :** Untuk mengecek apakah ipk pada indeks j lebih kecil dari ipk pada indeks idxMin jika lebih kecil maka idxMin akan dirubah valuenya menjadi index j.

## Percobaan 3: Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

```
1  void insertionSort() {
2      for (int i = 1; i < listMhs.length; i++) {
3          Mahasiswa temp = listMhs[i];
4          int j = i;
5          while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
6              listMhs[j] = listMhs[j - 1];
7              j--;
8          }
9          listMhs[j] = temp;
10     }
11 }
```

```
1  System.out.println("Data Mahasiswa setelah sorting asc berdasarkan IPK");
2  list.insertionSort();
3  list.tampil();
```

Output :

```
Data Mahasiswa setelah sorting asc berdasarkan IPK
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.00
=====
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.00
=====
Nama = Dompur
Tahun Masuk = 2018
Umur = 19
IPK = 3.50
=====
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
=====
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.00
=====
```

### Pertanyaan

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

```
1  void insertionSort() {
2      for (int i = 1; i < listMhs.length; i++) {
3          Mahasiswa temp = listMhs[i];
4          int j = i;
5          while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
6              listMhs[j] = listMhs[j - 1];
7              j--;
8          }
9          listMhs[j] = temp;
10     }
11 }
```

Output :

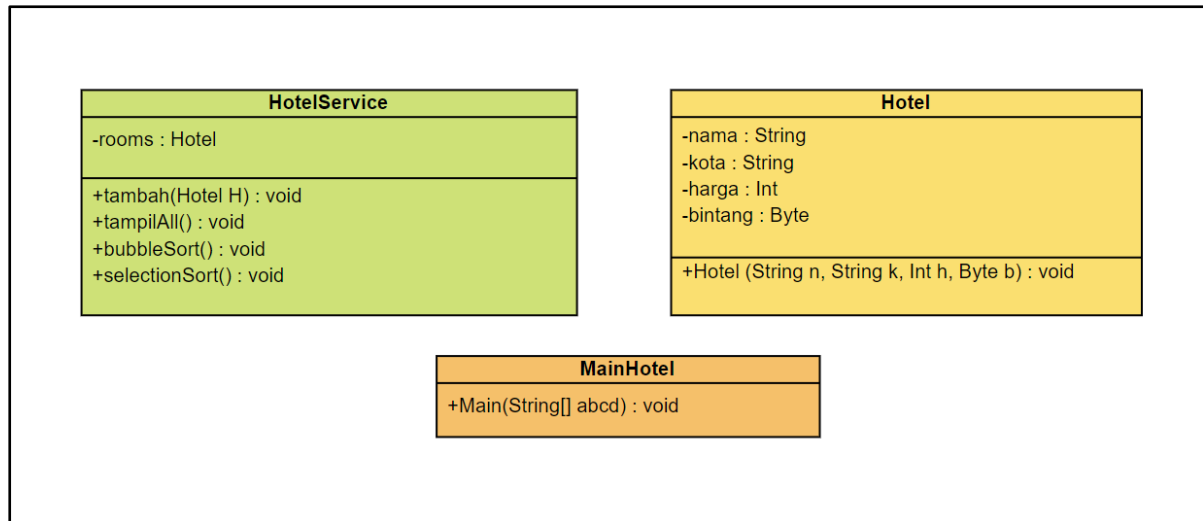
```
Data Mahasiswa setelah sorting asc berdasarkan IPK
Nama = Rara
Tahun Masuk = 2012
Umur = 19
IPK = 4.00
=====
Nama = Ummi
Tahun Masuk = 2019
Umur = 21
IPK = 3.75
=====
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.50
=====
Nama = Nusa
Tahun Masuk = 2017
Umur = 25
IPK = 3.00
=====
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.00
=====
```

### Latihan Praktikum

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

- a) Harga dimulai dari harga termurah ke harga tertinggi.
- b) 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.



Hotel.java

```
1  package jobsheet6;
2
3  public class Hotel {
4      String nama, kota;
5      int harga;
6      byte bintang;
7
8      Hotel(String n, String k, int h, byte b) {
9          nama = n;
10         kota = k;
11         harga = h;
12         bintang = b;
13     }
14
15 }
```

HotelService.java



```
1  package jobsheet6;
2
3  public class HotelService {
4      Hotel rooms[] = new Hotel[4];
5      int idx;
6
7      void tambah(Hotel h) {
8          if (idx <= rooms.length) {
9              rooms[idx] = h;
10             idx++;
11         } else {
12             System.out.println("Data sudah penuh !!");
13         }
14     }
15
16     void tampilAll() {
17         for (Hotel h : rooms) {
18             System.out.println("Nama Hotel: " + h.nama);
19             System.out.println("Kota: " + h.kota);
20             System.out.println("Harga: " + h.harga);
21             System.out.println("Bintang: " + h.bintang);
22             System.out.println("=====");
23         }
24     }
25
26     void bubbleSort() {
27         for (int i = 0; i < rooms.length - 1; i++) {
28             for (int j = 1; j < rooms.length - i; j++) {
29                 Hotel tmp = rooms[j];
30                 if (rooms[j].harga < rooms[j - 1].harga) {
31                     rooms[j] = rooms[j - 1];
32                     rooms[j - 1] = tmp;
33                 }
34             }
35         }
36     }
```

```

38     void selectionSort() {
39         for (int i = 0; i < rooms.length - 1; i++) {
40             int idxMin = i;
41             for (int j = i + 1; j < rooms.length; j++) {
42                 if (rooms[j].bintang > rooms[idxMin].bintang) {
43                     idxMin = j;
44                 }
45                 Hotel tmp = rooms[idxMin];
46                 rooms[idxMin] = rooms[i];
47                 rooms[i] = tmp;
48             }
49         }
50     }
51 }

```

MainHotel.java

```

1  package jobsheet6;
2
3  import java.util.Scanner;
4
5  public class MainHotel {
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8
9          HotelService li = new HotelService();
10         Hotel h1 = new Hotel("A", "Malang", 2_000_000, (byte) 5);
11         Hotel h2 = new Hotel("B", "Batu", 2_500_000, (byte) 1);
12         Hotel h3 = new Hotel("C", "Malang", 4_000_000, (byte) 2);
13         Hotel h4 = new Hotel("D", "Batu", 7_000_000, (byte) 4);
14
15         li.tambah(h1);
16         li.tambah(h2);
17         li.tambah(h3);
18         li.tambah(h4);
19
20         System.out.println("Menu Filter");
21         System.out.println("1. Harga\n2. Bintang");
22         System.out.print("Pilih: ");
23         int menu = sc.nextInt();

```

```

25     switch (menu) {
26         case 1:
27             System.out.println("\nData Hotel Berdasarkan Harga");
28             System.out.println("=====");
29             li.bubbleSort();
30             li.tampilAll();
31             break;
32         case 2:
33             System.out.println("\nData Hotel Berdasarkan Bintang");
34             System.out.println("=====");
35             li.selectionSort();
36             li.tampilAll();
37         default:
38             break;
39     }
40 }
41 }
42

```

Output Filter untuk Harga :

```

Menu Filter
1. Harga
2. Bintang
Pilih: 1

Data Hotel Berdasarkan Harga
=====
Nama Hotel: A
Kota: Malang
Harga: 20000000
Bintang: 5
=====
Nama Hotel: B
Kota: Batu
Harga: 25000000
Bintang: 1
=====
Nama Hotel: C
Kota: Malang
Harga: 40000000
Bintang: 2
=====
Nama Hotel: D
Kota: Batu
Harga: 70000000
Bintang: 4
=====

```

Output Filter untuk Bintang :

```

Menu Filter
1. Harga
2. Bintang
Pilih: 2

Data Hotel Berdasarkan Bintang
=====
Nama Hotel: A
Kota: Malang
Harga: 20000000
Bintang: 5
=====
Nama Hotel: D
Kota: Batu
Harga: 70000000
Bintang: 4
=====
Nama Hotel: C
Kota: Malang
Harga: 40000000
Bintang: 2
=====
Nama Hotel: B
Kota: Batu
Harga: 25000000
Bintang: 1
=====

```