



Name : Azaria Cindy Sahasika
 Number Id : 2341760169 / 06
 Class : 1G – Business Information System
 Lesson : Algorithm and Data Structure
 Material : Jobsheet 5
 Github Link : <https://github.com/azariacindy/algorithm-ds>

JOBSHEET - 5

SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

5.1 Tujuan Praktikum

Setelah melakukan praktikum ini diharapkan mahasiswa mampu:

- Mahasiswa mampu membuat algoritma searching bubble sort, selection sort dan insertion sort
- 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 berfungsi untuk membuat objek mahasiswa yang akan dimasukkan 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("Dompur", 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 =		Data mahasiswa setelah sorting desc berdasarkan ipk	
Nama	= Nusa	Nama	= Rara
Tahun Masuk	= 2017	Tahun Masuk	= 2012
Umur	= 25	Umur	= 19
IPK	= 3.0	IPK	= 4.0
-----		-----	
Nama	= Rara	Nama	= Ummi
Tahun Masuk	= 2012	Tahun Masuk	= 2019
Umur	= 19	Umur	= 21
IPK	= 4.0	IPK	= 3.75
-----		-----	
Nama	= Dompu	Nama	= Dompu
Tahun Masuk	= 2018	Tahun Masuk	= 2018
Umur	= 19	Umur	= 19
IPK	= 3.5	IPK	= 3.5
-----		-----	
Nama	= Abdul	Nama	= Nusa
Tahun Masuk	= 2017	Tahun Masuk	= 2017
Umur	= 23	Umur	= 25
IPK	= 2.0	IPK	= 3.0
-----		-----	
Nama	= Ummi	Nama	= Abdul
Tahun Masuk	= 2019	Tahun Masuk	= 2017
Umur	= 21	Umur	= 23
IPK	= 3.75	IPK	= 2.0
-----		-----	

Data Sebelum Diurutkan:		Data Setelah Diurutkan berdasarkan IPK:	
Name	: Azaria	Name	: Garcia
Year	: 2020	Year	: 2020
Age	: 20	Age	: 20
IPK	: 3.5	IPK	: 3.9
-----		-----	
Name	: Cindy	Name	: Sahasika
Year	: 2019	Year	: 2021
Age	: 21	Age	: 19
IPK	: 3.7	IPK	: 3.8
-----		-----	
Name	: Sahasika	Name	: Cindy
Year	: 2021	Year	: 2019
Age	: 19	Age	: 21
IPK	: 3.8	IPK	: 3.7
-----		-----	
Name	: Garcia	Name	: Catrine
Year	: 2020	Year	: 2023
Age	: 20	Age	: 18
IPK	: 3.9	IPK	: 3.6
-----		-----	
Name	: Catrine	Name	: Azaria
Year	: 2023	Year	: 2020
Age	: 18	Age	: 20
IPK	: 3.6	IPK	: 3.5
-----		-----	

5.2.3 Pertanyaan

1. Terdapat di method apakah proses bubble sort?

```
jobsheet5 > J dafMhsBpres06.java > ...
3 public class dafMhsBpres06 {
16 void bubbleSort(){
17     for (int i = 0; i < listMhs.length - 1; i++) {
18         for (int j = 0; j < listMhs.length - 1 - i; j++) {
19             if (listMhs[j].ipk < listMhs[j + 1].ipk) {
20                 mhs06 tmp = listMhs[j];
21                 listMhs[j] = listMhs[j + 1];
22                 listMhs[j + 1] = tmp;
23             }
24         }
25     }
26 }
```

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

```
if(listMhs[j].ipk > listMhs[j-1].ipk){
    //di bawah ini proses swap atau penukaran
    Mahasiswa tmp = listMhs[j];
    listMhs[j] = listMhs[j-1];
    listMhs[j-1] = tmp;
}
```

Untuk apakah proses tersebut?

- ➔ Kode diatas sedang melakukan proses perbandingan. Jika ipk mahasiswa 'j' kurang dari ipk mahasiswa 'j-1', maka akan melakukan swap nilai antara 'j' dan 'j+1'.

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?
 - ➔ 'i' berfungsi untuk melakukan looping selama masih ada elemen yang perlu di swap.
 - ➔ 'j' berfungsi untuk melakukan perbandingan dan swap nilai antara 2 elemen yang berdekatan.
- b. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?
 - ➔ Karena pada perulangan terakhir nilai 'i' akan bernilai ' $\text{listMhs.length} - 1$ ' dan perbandingan akan dimulai jika 'j' bernilai 0.
- c. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - ➔ Karena pada looping setiap kali urutan terjadi, jumlah perbandingan yang perlu dilakukan akan berkurang dari ' $n - 1$ ' ke ' $n - i$ ' ($n = \text{'listMhs'}$).
- d. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
 - ➔ Jika listMhs adalah 50, maka total looping 'i' akan berlangsung 49x, dan jumlah tahap bubble sort akan ada 49x.

5.3 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort

Waktu : 30 menit

Jika pada praktikum yang sebelumnya kita telah mengurutkan data mahasiswa 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?

➔ Sudah!

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	= 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 asc berdasarkan ipk	
		Nama	= Abdul
		Tahun Masuk	= 2017
		Umur	= 23
		IPK	= 2.0

		Nama	= Nusa
		Tahun Masuk	= 2017
		Umur	= 25
		IPK	= 3.0

		Nama	= Dompur
		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

Data Sebelum Diurutkan:			
Name	: Azaria		
Year	: 2020		
Age	: 20		
IPK	: 3.5		

Name	: Cindy		
Year	: 2019		
Age	: 21		
IPK	: 3.7		

Name	: Sahasika		
Year	: 2021		
Age	: 19		
IPK	: 3.8		

Name	: Garcia		
Year	: 2020		
Age	: 20		
IPK	: 3.9		

Name	: Catrine		
Year	: 2023		
Age	: 18		
IPK	: 3.6		

		Data Setelah Diurutkan (Selection Sort) berdasarkan IPK:	
		Name	: Azaria
		Year	: 2020
		Age	: 20
		IPK	: 3.5

		Name	: Catrine
		Year	: 2023
		Age	: 18
		IPK	: 3.6

		Name	: Cindy
		Year	: 2019
		Age	: 21
		IPK	: 3.7

		Name	: Sahasika
		Year	: 2021
		Age	: 19
		IPK	: 3.8

		Name	: Garcia
		Year	: 2020
		Age	: 20
		IPK	: 3.9

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!

```

int idxMin = i; // nilai min awal di simpan
for(int j = i + 1; j < listMhs.length; j++) { // dimulai dari 1 karena min sudah diketahui
    if(listMhs[j].ipk < listMhs[idxMin].ipk){ // melakukan perbandingan
        idxMin = j; // update nilai min
    }
}
    
```

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?

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

Data Sebelum Diurutkan:

Name : Azaria
Year : 2020
Age : 20
IPK : 3.5

Name : Cindy
Year : 2019
Age : 21
IPK : 3.7

Name : Sahasika
Year : 2021
Age : 19
IPK : 3.8

Name : Garcia
Year : 2020
Age : 20
IPK : 3.9

Name : Catrine
Year : 2023
Age : 18
IPK : 3.6

Data Setelah Diurutkan Asc (Insertion Sort) berdasarkan IPK:

Name : Azaria
Year : 2020
Age : 20
IPK : 3.5

Name : Catrine
Year : 2023
Age : 18
IPK : 3.6

Name : Cindy
Year : 2019
Age : 21
IPK : 3.7

Name : Sahasika
Year : 2021
Age : 19
IPK : 3.8

Name : Garcia
Year : 2020
Age : 20
IPK : 3.9

5.4.3 Pertanyaan

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



```
Data Setelah Diurutkan Desc (Insertion Sort) berdasarkan IPK:
Name : Garcia
Year : 2020
Age : 20
IPK : 3.9
-----
Name : Sahasika
Year : 2021
Age : 19
IPK : 3.8
-----
Name : Cindy
Year : 2019
Age : 21
IPK : 3.7
-----
Name : Catrine
Year : 2023
Age : 18
IPK : 3.6
-----
Name : Azaria
Year : 2020
Age : 20
IPK : 3.5
-----
```

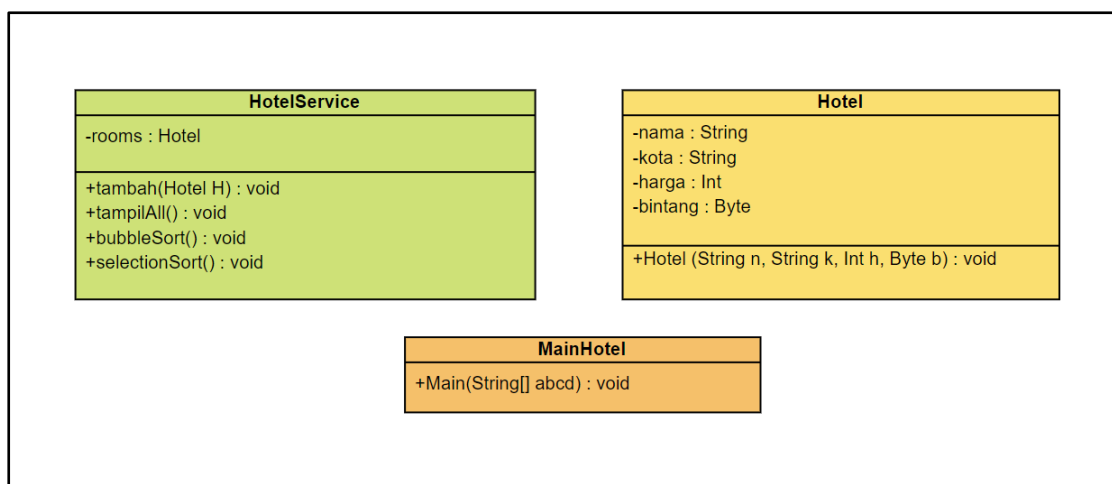
5.5 Latihan Praktikum

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**.





Daftar penginapan sebelum sorting berdasarkan harga:	Daftar penginapan sebelum sorting berdasarkan rating:
Name : Hotel 5 Star	Name : Guest House Asri
City : Jakarta	City : Malang
Price : 150	Price : 100
Rating : 4	Rating : 3
-----	-----
Name : Villa Unik	Name : Hotel 5 Star
City : Batu	City : Jakarta
Price : 200	Price : 150
Rating : 5	Rating : 4
-----	-----
Name : Guest House Asri	Name : Villa Unik
City : Malang	City : Batu
Price : 100	Price : 200
Rating : 3	Rating : 5
-----	-----
Name : Resort Mevvah	Name : Resort Mevvah
City : Bandung	City : Bandung
Price : 250	Price : 250
Rating : 4	Rating : 4
-----	-----
Daftar penginapan setelah sorting berdasarkan harga:	Daftar penginapan setelah sorting berdasarkan rating:
Name : Guest House Asri	Name : Villa Unik
City : Malang	City : Batu
Price : 100	Price : 200
Rating : 3	Rating : 5
-----	-----
Name : Hotel 5 Star	Name : Hotel 5 Star
City : Jakarta	City : Jakarta
Price : 150	Price : 150
Rating : 4	Rating : 4
-----	-----
Name : Villa Unik	Name : Resort Mevvah
City : Batu	City : Bandung
Price : 200	Price : 250
Rating : 5	Rating : 4
-----	-----
Name : Resort Mevvah	Name : Guest House Asri
City : Bandung	City : Malang
Price : 250	Price : 100
Rating : 4	Rating : 3
-----	-----



```

jobsheet5 > J travel06.java > ...
Click here to ask Blackbox to help you code faster
1 package jobsheet5;
2
3 public class travel06 {
4     String nama, kota;
5     int harga;
6     byte rating;
7
8     public travel06(String n, String k, int h, byte r) {
9         nama = n;
10        kota = k;
11        harga = h;
12        rating = r;
13    }
14
15    public String getName() {
16        return nama;
17    }
18
19    public int getHarga() {
20        return harga;
21    }
22
23    public byte getRating() {
24        return rating;
25    }
26
27    void show(){
28        System.out.println("Name : "+ nama);
29        System.out.println("City : "+ kota);
30        System.out.println("Price : "+ harga);
31        System.out.println("Rating : "+ rating);
32    }
33 }

```

```

jobsheet5 > J travelSort06.java > ...
Click here to ask Blackbox to help you code faster
1 package jobsheet5;
2
3 public class travelSort06 {
4     // Bubble Sort untuk sorting berdasarkan harga
5     public static void bubbleSortByPrice(travel06[] penginapan) {
6         int n = penginapan.length;
7         for (int i = 0; i < n - 1; i++) {
8             for (int j = 0; j < n - i - 1; j++) {
9                 if (penginapan[j].harga > penginapan[j + 1].harga) {
10                    travel06 temp = penginapan[j];
11                    penginapan[j] = penginapan[j + 1];
12                    penginapan[j + 1] = temp;
13                }
14            }
15        }
16    }
17
18    // Selection Sort untuk sorting berdasarkan rating
19    public static void selectionSortByRating(travel06[] penginapan) {
20        int n = penginapan.length;
21        for (int i = 0; i < n - 1; i++) {
22            int minIndex = i;
23            for (int j = i + 1; j < n; j++) {
24                if (penginapan[j].rating > penginapan[minIndex].rating) {
25                    minIndex = j;
26                }
27            }
28            travel06 temp = penginapan[minIndex];
29            penginapan[minIndex] = penginapan[i];
30            penginapan[i] = temp;
31        }
32    }
33
34    // Fungsi untuk menampilkan daftar penginapan
35    public static void printPenginapan(travel06[] penginapan) {
36        for (travel06 p : penginapan) {
37            p.show();
38            System.out.println(x: "-----");
39        }
40    }
41 }

```

```

jobsheet5 > J travelMain06.java > ...
Click here to ask Blackbox to help you code faster
1 package jobsheet5;
2
3 public class travelMain06 {
4     Run | Debug
5     public static void main(String[] args) {
6         travel06[] penginapan = {
7             new travel06(n:"Hotel 5 Star", k:"Jakarta", h:150, (byte) 4),
8             new travel06(n:"Villa Unik", k:"Batu", h:200, (byte) 5),
9             new travel06(n:"Guest House Asri", k:"Malang", h:100, (byte) 3),
10            new travel06(n:"Resort Mevvah", k:"Bandung", h:250, (byte) 4)
11        };
12
13        System.out.println(x:"Daftar penginapan sebelum sorting berdasarkan harga:");
14        travelSort06.printPenginapan(penginapan);
15
16        travelSort06.bubbleSortByPrice(penginapan);
17        System.out.println(x:"\nDaftar penginapan setelah sorting berdasarkan harga:");
18        travelSort06.printPenginapan(penginapan);
19
20        travelSort06.selectionSortByRating(penginapan);
21        System.out.println(x:"\nDaftar penginapan setelah sorting berdasarkan rating:");
22        travelSort06.printPenginapan(penginapan);
23    }
24 }

```