

JOB SHEET - 5

SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

Nama : Muhammad Abhinaya Zurfa

Kelas : SIB 1 C / 20

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 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 DaftarMahasiswaBerprestasi {
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("Dompus", 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      = Dompus
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 sebelum sorting:
Nama: Nusa
Tahun Masuk: 2017
Umur: 25
IPK: 3.0
-----
Nama: Rara
Tahun Masuk: 2012
Umur: 19
IPK: 4.0
-----
Nama: Dompus
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
-----
```

```
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
-----
PS D:\TUGAS KULIAH\SEMESTER 2\ALGORITMA DAN STRUKTUR I
```

5.2.3 Pertanyaan

1. Terdapat di method apakah proses bubble sort?

Terdapat pada method void bubbleSort() pada class daftar mahasiswa berprestasi

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?

Pada proses tersebut terdapat proses pemilihan menggunakan if yang digunakan untuk menukar 2 elemen, dimana terdapat kondisi jika (listMhs[j].ipk > listMhs[j-1].ipk) terpenuhi, maka selanjutnya dilakukan proses penukaran / swap mengurutkan elemen array listMhs berdasarkan IPK dalam urutan descending / dari besar ke kecil.

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 merupakan outer loop yang digunakan mengurangi jumlah elemen yang perlu diurutkan dengan setiap putaran
 - Perulangan j merupakan inner loop yang digunakan untuk perbandingan dan penukaran elemen yang sebenarnya dalam array pada setiap perulangan i
- b. Mengapa syarat dari perulangan i adalah i<listMhs.length-1 ?

Agar batas dari outer loop i mempunyai panjang elemen sejumlah Panjang array listMhs dikurangi 1 ketika melakukan perulangan dan memastikan bahwa perulangan tidak mengakses elemen array yang tidak ada, sehingga mencegah error



- c. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
Agar batas dalam inner loop j mempunyai Panjang elmen sejumlah Panjang array listMhs dikurangi i saat proses swap nilai dari array agar bisa urut
- d. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?

Perulangan i akan berlangsung selama 49 kali ($n-1$) karena pada setiap iterasi dari perulangan i, elemen terakhir dari array posisi nya dianggap sudah benar dan untuk tahap bubble sort jika Panjang listMhs 50 yaitu sebanyak Panjang dari array listMhs – $i(\text{listMhs.length} - i)$ sampai proses swap nilai dari elemen array sudah memenuhi kondisi terurut

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? Urut ascending

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      = Dompus
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 sebelum sorting:
Nama: Nusa
Tahun Masuk: 2017
Umur: 25
IPK: 3.0
-----
Nama: Rara
Tahun Masuk: 2012
Umur: 19
IPK: 4.0
-----
Nama: Dompus
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      = 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
-----
```

```
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
-----
```

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!

Pertama menginisialisasikan variable `idxMin` dengan nilai `i`, yang merupakan indeks awal dari sub array yang belum terurut, lalu for loop dijalankan dimulai dari `j=i+1` hingga akhir array `listMhs.length`, lalu dalam for ada kondisi if untuk membandingkan nilai ipk dari elemen yang dicek (`listMhs[j].ipk` dengan nilai ipk dari elemen dengan indeks `listMhs[idxMin].ipk`



Jika nilai ipk dari elemen yang sedang di cek lebih kecil dari nilai ip elemen $idxMin$, maka akan $==$ indeks j , setelah iterasi selesai $idxMin$ akan menunjukkan indeks dari elemen dengan nilai ipk terkecil dalam sub array yang belum terurut

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? Urut ascending

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 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 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.

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

```
    }  
    listMhs[j] = temp;  
  }  
}
```

```
-----  
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  
-----
```

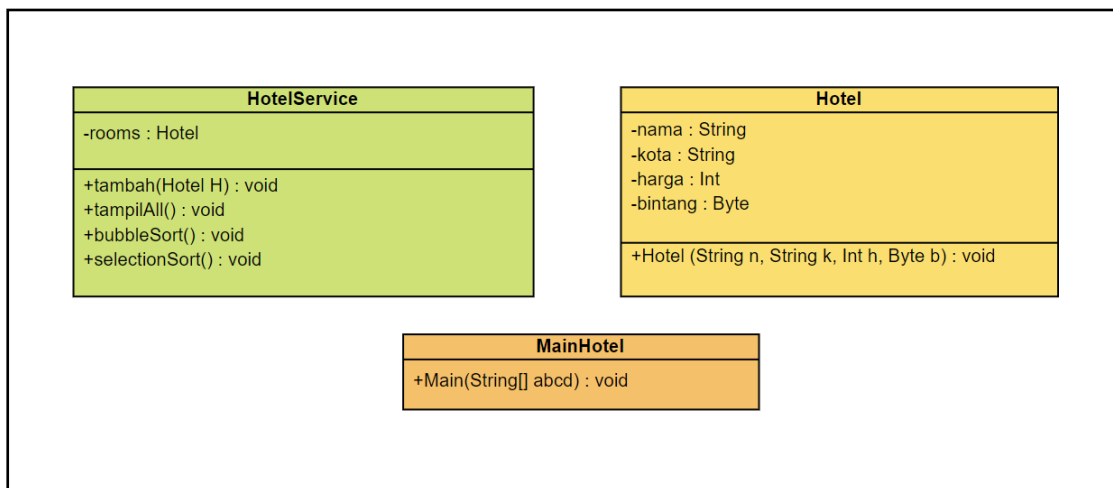
5.5 Latihan Praktikum

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**.



```

package jobsheet5;
/**
 * @author Abhinaya
 */
public class HotelService {
    Hotel rooms[] = new Hotel[5];
    int index;

    void tambah(Hotel h){
        if(index < rooms.length){
            rooms[index] = h;
            index++;
        }
    }

    void tampil(){
        for(Hotel h : rooms){
            h.tampilAll();
            System.out.println("=====
=");
        }
    }
}
    
```

```
// method sorting berdasarkan harga
void bubbleSort(){
    for(int i = 0; i < rooms.length-1; i++){
        for(int j = 1; j < rooms.length-i; j++){
            if(rooms[j].harga < rooms[j-1].harga){
                Hotel temp = rooms[j];
                rooms[j] = rooms[j-1];
                rooms[j-1] = temp;
            }
        }
    }
}

void selectionSort(){
    for(int i = 0; i < rooms.length-1; i++){
        int min = i;
        for(int j = i+1; j < rooms.length; j++){
            if(rooms[j].harga < rooms[min].harga){
                min = j;
            }
        }
        Hotel temp = rooms[min];
        rooms[min] = rooms[i];
        rooms[i] = temp;
    }
}

// method sorting berdasarkan bintang
void bubbleSortBintang(){
    for(int i = 0; i < rooms.length-1; i++){
        for(int j = 1; j < rooms.length-i; j++){
            if(rooms[j].bintang > rooms[j-1].bintang){
                Hotel temp = rooms[j];
                rooms[j] = rooms[j-1];
                rooms[j-1] = temp;
            }
        }
    }
}

void selectionSortBintang(){
    for(int i = 0; i < rooms.length-1; i++){
        int min = i;
        for(int j = i+1; j < rooms.length; j++){
            if(rooms[j].bintang > rooms[min].bintang){
                min = j;
            }
        }
    }
}
```



```

        Hotel temp = rooms[min];
        rooms[min] = rooms[i];
        rooms[i] = temp;
    }
}
}

```

```

package jobsheet5;
/**
 * @author Abhinaya
 */
public class MainHotel {
    public static void main(String[] args) {
        HotelService list = new HotelService();
        Hotel h1 = new Hotel();
        h1.Hotel("Aston", "Bali", 500000, (byte) 5);
        Hotel h2 = new Hotel();
        h2.Hotel("Hilton", "Jakarta", 750000, (byte) 5);
        Hotel h3 = new Hotel();
        h3.Hotel("Ibis", "Bandung", 300000, (byte) 3);
        Hotel h4 = new Hotel();
        h4.Hotel("Santika", "Yogyakarta", 400000, (byte) 4);
        Hotel h5 = new Hotel();
        h5.Hotel("Novotel", "Surabaya", 600000, (byte) 4);

        list.tambah(h1);
        list.tambah(h2);
        list.tambah(h3);
        list.tambah(h4);
        list.tambah(h5);

        System.out.println("-----");
        System.out.println("----- Data Hotel Sebelum Sorting -----");
        System.out.println("-----");
        list.tampil();

        System.out.println("-----");
        System.out.println("----- Data Hotel Setelah Sorting -----");
        System.out.println("----- Berdasarkan Harga (Bubble Sort) -----");
        System.out.println("-----");
        list.bubbleSort();
        list.tampil();

        System.out.println("-----");
        System.out.println("----- Data Hotel Setelah Sorting -----");
        System.out.println("----- Berdasarkan Harga (Selection Sort) -----");
        System.out.println("-----");
        list.selectionSort();
    }
}

```



```

        list.tampil();

        System.out.println("-----");
        System.out.println("----- Data Hotel Setelah Sorting -----");
        System.out.println("----- Berdasarkan Bintang (Bubble Sort) -----");
        System.out.println("-----");
        list.bubbleSortBintang();
        list.tampil();

        System.out.println("-----");
        System.out.println("----- Data Hotel Setelah Sorting -----");
        System.out.println("--- Berdasarkan Bintang (Selection Sort) ---");
        System.out.println("-----");
        list.selectionSortBintang();
        list.tampil();
    }
}

```

```

package jobsheet5;
/**
 * @author Abhinaya
 */
public class Hotel {
    String nama, kota;
    int harga;
    byte bintang;

    void Hotel(String n, String k, int h, byte b) {
        nama = n;
        kota = k;
        harga = h;
        bintang = b;
    }
    void tampilAll(){
        System.out.println("Nama Hotel\t: " + nama);
        System.out.println("Kota\t\t: " + kota);
        System.out.println("Harga\t\t: " + harga);
        System.out.println("Bintang\t\t: " + bintang);
    }
}

```




```

----- Data Hotel Sebelum Sorting -----
Nama Hotel      : Aston
Kota            : Bali
Harga          : 500000
Bintang        : 5
=====
Nama Hotel      : Hilton
Kota            : Jakarta
Harga          : 750000
Bintang        : 5
=====
Nama Hotel      : Ibis
Kota            : Bandung
Harga          : 300000
Bintang        : 3
=====
Nama Hotel      : Santika
Kota            : Yogyakarta
Harga          : 400000
Bintang        : 4
=====
Nama Hotel      : Novotel
Kota            : Surabaya
Harga          : 600000
Bintang        : 4
=====
    
```

```

----- Data Hotel Setelah Sorting -----
----- Berdasarkan Harga (Bubble Sort) -----
Nama Hotel      : Ibis
Kota            : Bandung
Harga          : 300000
Bintang        : 3
=====
Nama Hotel      : Santika
Kota            : Yogyakarta
Harga          : 400000
Bintang        : 4
=====
Nama Hotel      : Aston
Kota            : Bali
Harga          : 500000
Bintang        : 5
=====
Nama Hotel      : Novotel
Kota            : Surabaya
Harga          : 600000
Bintang        : 4
=====
Nama Hotel      : Hilton
Kota            : Jakarta
Harga          : 750000
Bintang        : 5
=====
    
```

```

----- Data Hotel Setelah Sorting -----
----- Berdasarkan Harga (Selection Sort) -----
Nama Hotel      : Ibis
Kota            : Bandung
Harga          : 300000
Bintang        : 3
=====
Nama Hotel      : Santika
Kota            : Yogyakarta
Harga          : 400000
Bintang        : 4
=====
Nama Hotel      : Aston
Kota            : Bali
Harga          : 500000
Bintang        : 5
=====
Nama Hotel      : Novotel
Kota            : Surabaya
Harga          : 600000
Bintang        : 4
=====
Nama Hotel      : Hilton
Kota            : Jakarta
Harga          : 750000
Bintang        : 5
=====
    
```

```

----- Data Hotel Setelah Sorting -----
----- Berdasarkan Bintang (Selection Sort) -----
Nama Hotel      : Aston
Kota            : Bali
Harga          : 500000
Bintang        : 5
=====
Nama Hotel      : Hilton
Kota            : Jakarta
Harga          : 750000
Bintang        : 5
=====
Nama Hotel      : Santika
Kota            : Yogyakarta
Harga          : 400000
Bintang        : 4
=====
Nama Hotel      : Novotel
Kota            : Surabaya
Harga          : 600000
Bintang        : 4
=====
Nama Hotel      : Ibis
Kota            : Bandung
Harga          : 300000
Bintang        : 3
=====
    
```

```

----- Data Hotel Setelah Sorting -----
----- Berdasarkan Bintang (Bubble Sort) -----
Nama Hotel      : Aston
Kota            : Bali
Harga          : 500000
Bintang        : 5
=====
Nama Hotel      : Hilton
Kota            : Jakarta
Harga          : 750000
Bintang        : 5
=====
Nama Hotel      : Santika
Kota            : Yogyakarta
Harga          : 400000
Bintang        : 4
=====
Nama Hotel      : Novotel
Kota            : Surabaya
Harga          : 600000
Bintang        : 4
=====
Nama Hotel      : Ibis
Kota            : Bandung
Harga          : 300000
Bintang        : 3
=====
    
```