

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022

Praktikan

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi [harus diupdate setiap ada perubahan]

MENGURUTKAN DATA MAHASISWA BERDASARKAN IPK MENGGUNAKAN BUBBLE SORT	4
LANGKAH 1.....	4
LANGKAH 2.....	4
LANGKAH 3.....	4
LANGKAH 4.....	5
LANGKAH 5.....	5
LANGKAH 6.....	6
LANGKAH 7.....	6
LANGKAH 8.....	6
LANGKAH 9.....	7
VERIFIKASI HASIL PERCOBAAN	8
<i>Pertanyaan.....</i>	9
<i>Jawaban.....</i>	9
MENGURUTKAN DATA MAHASISWA BERDASARKAN IPK MENGGUNAKAN SELECTION SORT	11
LANGKAH 1.....	11
LANGKAH 2.....	11
LANGKAH 3.....	11
VERIFIKASI HASIL PERCOBAAN	11
<i>Pertanyaan.....</i>	12
<i>Jawaban.....</i>	13
MENGURUTKAN DATA MAHASISWA BERDASARKAN IPK MENGGUNAKAN INSERTION SORT	14
LANGKAH 1.....	14
LANGKAH 2.....	14
LANGKAH 3.....	14
VERIFIKASI HASIL PERCOBAAN	14
<i>Pertanyaan.....</i>	16
<i>Jawaban.....</i>	16
LATIHAN PRAKTIKUM	18
JAWABAN	19
<i>Source code class MainTiketRidwan</i>	<i>19</i>
<i>Source code class TiketRidwan.....</i>	<i>20</i>
<i>Source code class TiketServiceRidwan</i>	<i>21</i>

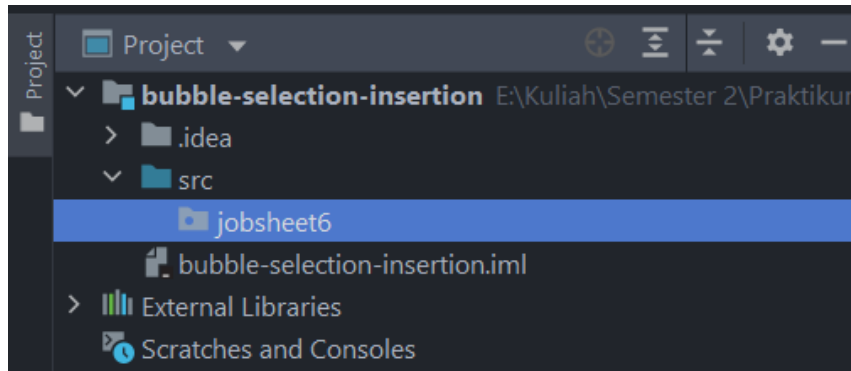


<i>Ouput</i>	22
--------------------	----

Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort

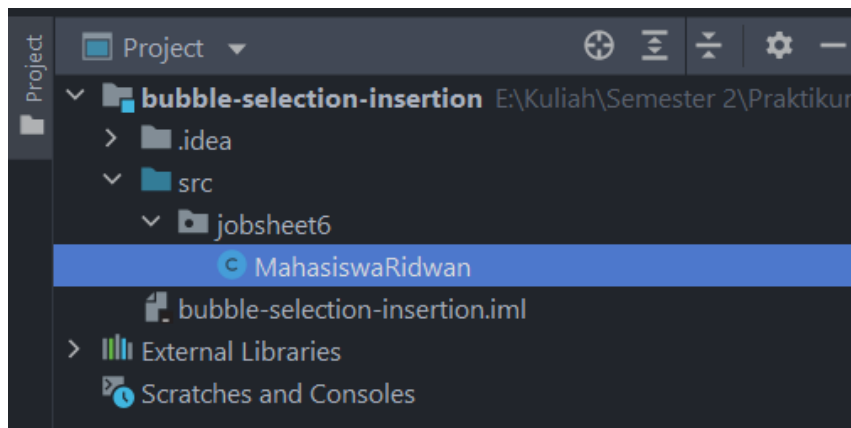
Langkah 1

Buat project baru dengan nama “bubble-selection-insertion”, kemudian buat package dengan nama “jobsheet6”.



Langkah 2

Buatlah sebuah class dengan nama Mahasiswa



Langkah 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

```
MahasiswaRidwan.java x
1  package jobsheet6;
2
3  public class MahasiswaRidwan {
4      String nama;
5      int thnMasuk, umur;
6      double ipk;
7
8      MahasiswaRidwan(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 }
22
```

Langkah 4

Buat class DaftarMahasiswaBerprestasi seperti di bawah ini!

```
MahasiswaRidwan.java x  DaftarMahasiswaBerprestasiRidwan.java x
1  package jobsheet6;
2
3  public class DaftarMahasiswaBerprestasiRidwan {
4      MahasiswaRidwan listMhs[] = new MahasiswaRidwan[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 }
13
```

Langkah 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(MahasiswaRidwan m) {
9          if (idx < listMhs.length) {
10             listMhs[idx] = m;
11             idx++;
12          } else {
13             System.out.println("Data sudah penuh!!!!");
14          }
15      }
```

Langkah 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 (MahasiswaRidwan m : listMhs) {
20             m.tampil();
21             System.out.println("-----");
22          }
23      }
```

Langkah 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                      MahasiswaRidwan tmp = listMhs[j];
32                      listMhs[j] = listMhs[j-1];
33                      listMhs[j-1] = tmp;
34                  }
35              }
36          }
37      }
```

Langkah 8

Buat class Main dan didalamnya buat method main() seperti di bawah ini!

```
MahasiswaRidwan.java x DaftarMahasiswaBerprestasiRidwan.java x MainRidwan.java
package jobsheet6;

public class MainRidwan {
    public static void main(String[] args) {
    }
}
```

Langkah 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 dimasukkan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.

```
MainRidwan.java x MahasiswaRidwan.java x DaftarMahasiswaBerprestasiRidwan.java x
1 package jobsheet6;
2
3 public class MainRidwan {
4     public static void main(String[] args) {
5         DaftarMahasiswaBerprestasiRidwan list = new DaftarMahasiswaBerprestasiRidwan();
6         MahasiswaRidwan m1 = new MahasiswaRidwan( n: "Nusa", t: 2017, u: 25, i: 3);
7         MahasiswaRidwan m2 = new MahasiswaRidwan( n: "Rara", t: 2012, u: 19, i: 4);
8         MahasiswaRidwan m3 = new MahasiswaRidwan( n: "Dompur", t: 2018, u: 19, i: 3.5);
9         MahasiswaRidwan m4 = new MahasiswaRidwan( n: "Abdul", t: 2017, u: 23, i: 2);
10        MahasiswaRidwan m5 = new MahasiswaRidwan( n: "Ummi", t: 2019, u: 21, i: 3.75);
11
12        list.tambah(m1);
13        list.tambah(m2);
14        list.tambah(m3);
15        list.tambah(m4);
16        list.tambah(m5);
17
18        System.out.println("Data mahasiswa sebelum sorting = ");
19        list.tampil();
20
21        System.out.println("Data mahasiswa setelah sorting desc berdasarkan ipk");
22        list.bubbleSort();
23        list.tampil();
24    }
25 }
```

Verifikasi hasil percobaan

```
Run: MainRidwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.
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 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
-----
```


Pertanyaan

1. Terdapat di method apakah proses bubble sort?
2. Terdapat di method apakah proses selection sort?
3. Apakah yang dimaksud proses swap? Tuliskan potongan program untuk melakukan proses swap tersebut!
4. 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?

5. 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?
- b) Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length}-1$?
- c) Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length}-i$?
- d) Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Jawaban

1. Proses bubble sort ada pada method bubbleSort()
2. Pada program diatas tidak ada proses yang menggunakan selection sort namun menggunakan bubble sort
3. Proses swap adalah proses pertukaran posisi dari elemen array.

Proses itu terjadi pada potongan baris berikut:

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

4. Pada baris program itu akan dilakukan pengecekan apakah nilai dari element array index ke j lebih besar dari element array index ke j-1. Jika iya maka akan dilakukan swap atau pertukaran posisi dan jika tidak maka akan dilanjutkan ke proses selanjutnya.
5. Penjelasan potongan program bubble sort
 - a) Perulangan i digunakan sebagai outerloop yang akan digunakan sebagai perulangan ketika proses swap sudah sampai ujung nilai array yang diinginkan, sedangkan untuk perulangan j digunakan untuk proses perulangan swap itu sendiri.

- b) Karena proses perulangan outerloop harus mengulang sebanyak panjang array dikurangi 1
- c) Karena proses perulangan innerloop harus berkurang sebanyak nilai iterasi I karena jika selesai satu iterasi outerloop maka nilai array paling akhir tidak dibandingkan lagi.
- d) Proses perulangan I akan diulang sebanyak 49 kali dan proses perulangan j akan diulang sebanyak 1225 kali

Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort

Langkah 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         MahasiswaRidwan tmp = listMhs[idxMin];
50         listMhs[idxMin] = listMhs[i];
51         listMhs[i] = tmp;
52     }
53 }
```

Langkah 2

Setelah itu, buka kembali class `Main`, dan di dalam method `main()` tambahkan baris program untuk memanggil method `selectionSort()` tersebut!

```
25 System.out.println("Data mahasiswa setelah sorting asc berdasarkan ipk");
26 list.selectionSort();
27 list.tampil();
```

Langkah 3

Coba jalankan kembali class `Main`, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

Verifikasi hasil percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

```
Run: MainRidwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.
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 = Dampu
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 = Dampu
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!

Jawaban

1. Proses tersebut pertama-tama menentukan nilai index minimal yaitu nilai I kemudian melakukan perulangan untuk membandingkan nilai element listMhs index ke j dengan index ke idxMin tadi yang dijadikan pembanding jika hasilnya nilai index j lebih kecil maka nilai j akan menjadi idxMin yang baru.

Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

Langkah 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.

```
55 //setelah ini tuliskan method insertionSort()
56 void insertionSort() {
57     for (int i = 1; i < listMhs.length; i++) {
58         MahasiswaRidwan temp = listMhs[i];
59         int j = i;
60         while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
61             listMhs[j] = listMhs[j-1];
62             j--;
63         }
64         listMhs[j] = temp;
65     }
66 }
```

Langkah 2

Setelah itu, buka kembali class `Main`, dan di dalam method `main()` tambahkan baris program untuk memanggil method `insertionSort()` tersebut!

```
29 System.out.println("Data mahasiswa setelah sorting asc berdasarkan ipk");
30 list.insertionSort();
31 list.tampil();
```

Langkah 3

Coba jalankan kembali class `Main`, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

Verifikasi hasil percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

```
Run: MainRidwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.
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
-----
```

Pertanyaan

Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara ascending atau decending, anda dapat melakukannya dengan menambahkan parameter pada pemanggilan fungsi insertionSort.

Jawaban

Source code class Main

```
29      System.out.println("Data mahasiswa setelah sorting asc berdasarkan ipk");
30      list.insertionSort( asc: true);
31      list.tampil();
32
33      System.out.println("Data mahasiswa setelah sorting desc berdasarkan ipk");
34      list.insertionSort( asc: false);
35      list.tampil();|
```

Source code class DaftarMahasiswaBerprestasi

```
55      //setelah ini tuliskan method insertionSort()
56      void insertionSort(boolean asc) {
57          for (int i = 1; i < listMhs.length; i++) {
58              MahasiswaRidwan temp = listMhs[i];
59              int j = i;
60              if (asc) {
61                  while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
62                      listMhs[j] = listMhs[j-1];
63                      j--;
64                  }
65              } else {
66                  while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
67                      listMhs[j] = listMhs[j-1];
68                      j--;
69                  }
70              }
71              listMhs[j] = temp;
72          }
73      }
```

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
```

```
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
```

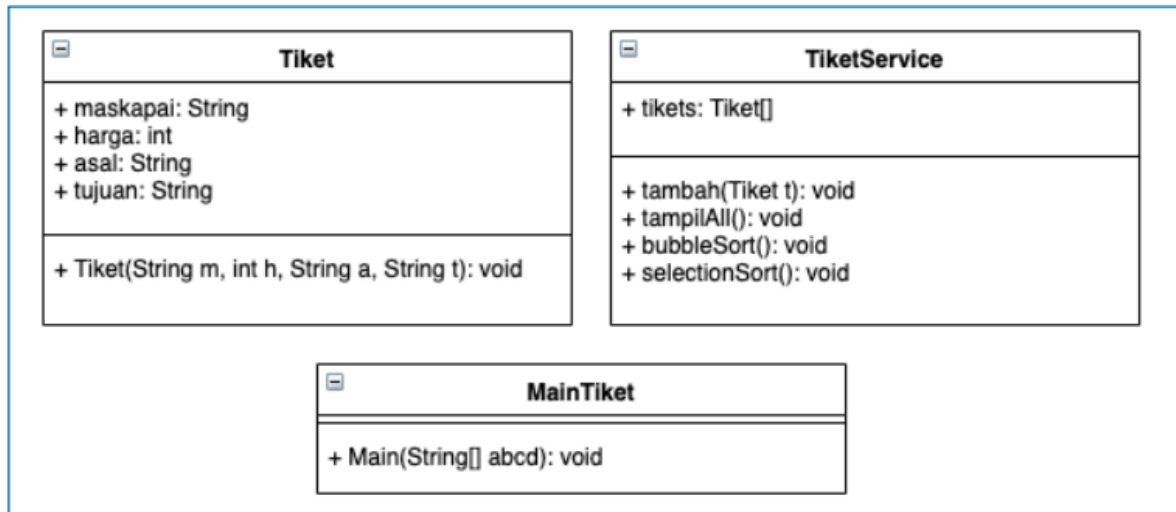
```
-----
Nama = Dompu
Tahun Masuk = 2018
Umur = 19
IPK = 3.5
```

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

```
-----
Nama = Abdul
Tahun Masuk = 2017
Umur = 23
IPK = 2.0
```

Latihan Praktikum

Sebuah yang bergerak dalam bidang penjualan tiket pesawat sedang mengembangkan backend untuk sistem pemesanan tiket, salah satu fiturnya adalah menampilkan daftar tiket yang tersedia berdasarkan pilihan filter yang diinginkan user. Daftar tiket ini harus dapat di sorting berdasarkan harga dimulai dari harga termurah ke harga tertinggi. Implementasikanlah class diagram berikut ini kedalam bahasa pemrograman java kemudian buatlah proses sorting data untuk harga tiket menggunakan algoritma bubble sort dan selection sort.



Jawaban

Source code class MainTiketRidwan

```

TiketRidwan.java x TiketServiceRidwan.java x MainTiketRidwan.java x
1 package LatihanPraktikumRidwan;
2
3 import java.util.Scanner;
4
5 public class MainTiketRidwan {
6     public static void main(String[] args) {
7         System.out.println("Selamat datang kedalam sistem penjualan tiket kami");
8
9         TiketServiceRidwan daftarTiket = new TiketServiceRidwan();
10        byte menu;
11        Scanner input = new Scanner(System.in);
12        TiketRidwan tiket1 = new TiketRidwan(m: "Super Air Jet", h: 470_000, a: "Surabaya", t: "Jakarta");
13        TiketRidwan tiket2 = new TiketRidwan(m: "Garuda Indonesia", h: 980_000, a: "Surabaya", t: "Jakarta");
14        TiketRidwan tiket3 = new TiketRidwan(m: "Batik Air", h: 680_000, a: "Surabaya", t: "Jakarta");
15        TiketRidwan tiket4 = new TiketRidwan(m: "Citilink Indonesia", h: 900_000, a: "Surabaya", t: "Jakarta");
16        TiketRidwan tiket5 = new TiketRidwan(m: "Lion", h: 480_000, a: "Surabaya", t: "Jakarta");
17
18        daftarTiket.tambah(tiket1);
19        daftarTiket.tambah(tiket2);
20        daftarTiket.tambah(tiket3);
21        daftarTiket.tambah(tiket4);
22        daftarTiket.tambah(tiket5);
23
24        System.out.println("Silahkan pilih filter daftar tiket berdasarkan harga : ");
25        System.out.println("[1] termurah");
26        System.out.println("[2] termahal");
27        System.out.println("=====");
28        System.out.print("Masukkan pilihan anda : ");
29        menu = input.nextByte();
30        System.out.println();
31
32        switch (menu) {
33            case 1 -> {
34                daftarTiket.bubbleSort();
35                System.out.println("==== Daftar tiket diurutkan dari termurah ===");
36                daftarTiket.tampilAll();
37            }
38            case 2 -> {
39                daftarTiket.selectionSort();
40                System.out.println("==== Daftar tiket diurutkan dari termahal ===");
41                daftarTiket.tampilAll();
42            }
43            default -> System.out.println("Inputan salah!");
44        }
45    }
46 }

```

Source code class TiketRidwan

```
TiketRidwan.java x TiketServiceRidwan.java x MainTiketRidwan.java x
1 package LatihanPraktikumRidwan;
2
3 public class TiketRidwan {
4     String maskapai;
5     int harga;
6     String asal;
7     String tujuan;
8
9     TiketRidwan(String m, int h, String a, String t) {
10         maskapai = m;
11         harga = h;
12         asal = a;
13         tujuan = t;
14     }
15
16     void tampil(int nomor) {
17         System.out.println("||\t\t\t\t Tiket " + nomor + " \t\t\t\t ||");
18         System.out.println("--> Maskapai = " + maskapai);
19         System.out.println("--> Harga = Rp." + harga);
20         System.out.println("--> Asal = " + asal);
21         System.out.println("--> Tujuan = " + tujuan);
22         System.out.println("=====");
23     }
24 }
```

Source code class TiketServiceRidwan

```

TiketRidwan.java x TiketServiceRidwan.java x MainTiketRidwan.java x
1  package LatihanPraktikumRidwan;
2
3  public class TiketServiceRidwan {
4      TiketRidwan[] tickets = new TiketRidwan[5];
5      int count;
6
7      void tambah(TiketRidwan t) {
8          if (count > tickets.length) {
9              System.out.println("Jumlah tiket sudah maksimal");
10         } else {
11             tickets[count] = t;
12             count++;
13         }
14     }
15
16     void tampilAll() {
17         int nomor = 1;
18         for (TiketRidwan t : tickets) {
19             t.tampil(nomor);
20             nomor++;
21         }
22     }
23 }
24
25 void bubbleSort() {
26     for (int i = 0; i < tickets.length - 1; i++) {
27         for (int j = 0; j < tickets.length - 1 - i; j++) {
28             if (tickets[j].harga > tickets[j + 1].harga) {
29                 TiketRidwan tmp = tickets[j];
30                 tickets[j] = tickets[j + 1];
31                 tickets[j + 1] = tmp;
32             }
33         }
34     }
35 }
36
37 void selectionSort() {
38     for (int i = 0; i < tickets.length - 1; i++) {
39         int indexMax = i;
40         for (int j = 1 + i; j < tickets.length; j++) {
41             if (tickets[indexMax].harga < tickets[j].harga) {
42                 indexMax = j;
43             }
44         }
45         TiketRidwan tmp = tickets[indexMax];
46         tickets[indexMax] = tickets[i];
47         tickets[i] = tmp;
48     }
49 }
50 }

```

Ouput

```
Run: MainTiketRidwan x "C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
Selamat datang kedalam sistem penjualan tiket kami
Silahkan pilih filter daftar tiket berdasarkan harga :
[1] termurah
[2] termahal
=====
Masukkan pilihan anda : 1

==== Daftar tiket diurutkan dari termurah ====
||          Tiket 1          ||
--> Maskapai = Super Air Jet
--> Harga    = Rp.470000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
=====
||          Tiket 2          ||
--> Maskapai = Lion
--> Harga    = Rp.480000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
=====
||          Tiket 3          ||
--> Maskapai = Batik Air
--> Harga    = Rp.680000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
=====
||          Tiket 4          ||
--> Maskapai = Citilink Indonesia
--> Harga    = Rp.900000
--> Asal     = Surabaya
--> Tujuan   = Jakarta

Run: MainTiketRidwan x "C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
Selamat datang kedalam sistem penjualan tiket kami
Silahkan pilih filter daftar tiket berdasarkan harga :
[1] termurah
[2] termahal
=====
Masukkan pilihan anda : 2

==== Daftar tiket diurutkan dari termahal ====
||          Tiket 1          ||
--> Maskapai = Garuda Indonesia
--> Harga    = Rp.980000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
=====
||          Tiket 2          ||
--> Maskapai = Citilink Indonesia
--> Harga    = Rp.900000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
=====
||          Tiket 3          ||
--> Maskapai = Batik Air
--> Harga    = Rp.680000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
=====
||          Tiket 4          ||
--> Maskapai = Lion
--> Harga    = Rp.480000
--> Asal     = Surabaya
--> Tujuan   = Jakarta
```