

LAPORAN TUGAS PRAKTIKUM
JOBSHEET 5
SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)



Disusun Oleh :
DHANISA PUTRI MASHILFA
NIM. 2341720212
TI-1E/07

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No. 9 Jatimulyo, Kecamatan Lowokwaru, Jatimulyo, Kec. Lowokwaru,
Kota Malang, Jawa Timur 6514

5.2 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort

5.2.1 Langkah - Langkah Praktikum Percobaan

5.2.2 Hasil Praktikum Percobaan

<pre>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 : Dombu 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</pre>	<pre>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 : Dombu 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 -----</pre>
---	--

5.2.3 Pertanyaan

1. Terdapat di method apakah proses bubble sort?
✧ Terdapat pada **DaftarMahasiswaBerprestasi**
2. Di dalam method bubbleSort(), terdapat baris program seperti di bawah ini:
Untuk apakah proses tersebut?

```
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 | }
```

if (listMhs[j].ipk > listMhs[j-1].ipk) { memeriksa nilai dari elemen j dan dari elemen j-1, jika nilai lebih besar maka akan ditukar posisi elemennya. Mahasiswa tmp = listMhs[j];

variable **tmp** digunakan sementara untuk menyimpan elemen index **j**. **listMhs[j] = listMhs[j-1]**; memindahkan elemen dari indeks **j - 1** ke indeks **j**. **listMhs[j-1] = tmp**; elemen yang disimpan sementara di **tmp** dipindahkan ke indeks **j-1**. elemen nilai **ipk** yang lebih besar akan naik atau bergerak ke posisi yang lebih tinggi dalam array.

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++){
```

- Apakah perbedaan antara kegunaan perulangan **i** dan perulangan **j**?
 - ✧ variable **i** sebagai indeks luar untuk perulangan terluar, sedangkan variable **j** digunakan indeks dalam perulangan dalam untuk perulangan terdalam.
- Mengapa syarat dari perulangan **j** adalah **j<listMhs.length-1**?
 - ✧ Sebagai penunjuk jumlah elemen yang sudah berurutan dari ujung array, setiap iterasi dari perulangan luar menghasilkan satu elemen yang berurutan di ujung array.
- Mengapa syarat dari perulangan **j** adalah **j<listMhs.length-i** ?
 - ✧ Untuk membandingkan elemen sampai dengan **listMhs.length - i - 1**
- Jika banyak data di dalam **listMhs** adalah 0, maka berapakah perulangan **i** akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
 - ✧ Jika data dari **listMhs** adalah 0, maka perulangannya akan tetap 0 dan kondisi **i > listMhs.length - 1** tidak akan pernah terpenuhi karena **listMhs.length - 1** juga akan bernilai 0

5.3 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort

5.3.1 Langkah - Langkah Praktikum Percobaan

5.3.2 Hasil Praktikum Percobaan

```
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 : Dombu
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      : Dombu
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

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!

untuk menentukan elemen terkecil yang belum diurutkan pada ipk, fungsi dari **int idxMin = i** untuk melacak index dengan nilai terkecil, **for (int j = i+1; j < listMhs.length; j++)** { untuk memeriksa setiap elemen yang belum diurutkan setelah

elemen yang sedang di proses, **if (listMhs[j].ipk < listMhs[idxMin].ipk) {** untuk memeriksa nilai ipk dari elemen yang sedang diproses apakah lebih kecil dari nilai ipk dari elemen **idxMin**, **idxMin = j** apabila nilai ipk dari elemen yang sedang diproses lebih kecil nilai ipk dari elemen **idxMin**, nilai itu akan diperbarui menjadi **j** yang menandakan bahwa elemen index **j** memiliki nilai ipk terkecil yang baru ditemukan.

5.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

5.4.1 Langkah - Langkah Praktikum Percobaan

5.4.2 Hasil Praktikum Percobaan

Data Mahasiswa Sebelum Sorting :	Data Mahasiswa Setelah Sorting ASC Berdasarkan IPK
Nama : Nusa Tahun Masuk : 2017 Umur : 25 IPK : 3.0	Nama : Abdul Tahun Masuk : 2017 Umur : 23 IPK : 2.0
----- Nama : Rara Tahun Masuk : 2012 Umur : 19 IPK : 4.0	----- Nama : Nusa Tahun Masuk : 2017 Umur : 25 IPK : 3.0
----- Nama : Dombu Tahun Masuk : 2018 Umur : 19 IPK : 3.5	----- Nama : Dombu 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
----- Nama : Ummi Tahun Masuk : 2019 Umur : 21 IPK : 3.75	----- Nama : Rara Tahun Masuk : 2012 Umur : 19 IPK : 4.0

5.4.3 Pertanyaan

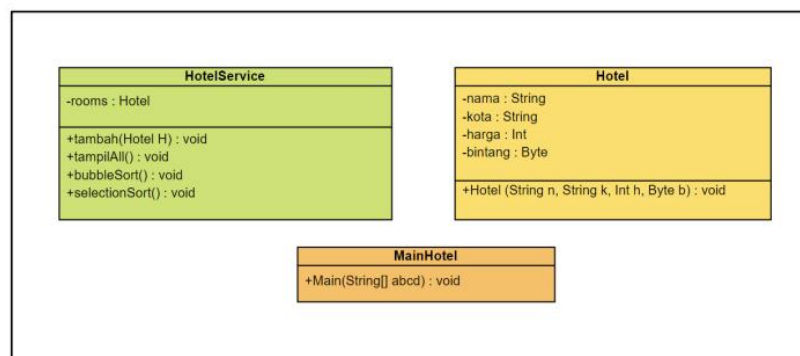
1. Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.

Data Mahasiswa Sebelum Sorting :	Data Mahasiswa Setelah Sorting ASC Berdasarkan IPK
Nama : Nusa Tahun Masuk : 2017 Umur : 25 IPK : 3.0	Nama : Rara Tahun Masuk : 2012 Umur : 19 IPK : 4.0
-----	-----
Nama : Rara Tahun Masuk : 2012 Umur : 19 IPK : 4.0	Nama : Ummi Tahun Masuk : 2019 Umur : 21 IPK : 3.75
-----	-----
Nama : Dombu Tahun Masuk : 2018 Umur : 19 IPK : 3.5	Nama : Dombu Tahun Masuk : 2018 Umur : 19 IPK : 3.5
-----	-----
Nama : Abdul Tahun Masuk : 2017 Umur : 23 IPK : 2.0	Nama : Nusa Tahun Masuk : 2017 Umur : 25 IPK : 3.0
-----	-----
Nama : Ummi Tahun Masuk : 2019 Umur : 21 IPK : 3.75	Nama : Abdul Tahun Masuk : 2017 Umur : 23 IPK : 2.0
-----	-----

5.5 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.

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



```
+-----+
| HOTEL RATING TERTINGGI |
+-----+

Hotel REMANG
Kota Saranjana
Harga Rp.400000
Rating Bintang5
-----
Hotel KAKTUS
Kota Saranjana
Harga Rp.350000
Rating Bintang4
-----
Hotel JINGGA
Kota Saranjana
Harga Rp.500000
Rating Bintang4
-----
Hotel BIRU
Kota Saranjana
Harga Rp.1000000
Rating Bintang3
-----
Hotel MAWAR
Kota Saranjana
Harga Rp.150000
Rating Bintang2
-----
```

```
+-----+
| HOTEL HARGA AFFORDEBLE |
+-----+

Hotel MAWAR
Kota Saranjana
Harga Rp.150000
Rating Bintang2
-----
Hotel KAKTUS
Kota Saranjana
Harga Rp.350000
Rating Bintang4
-----
Hotel REMANG
Kota Saranjana
Harga Rp.400000
Rating Bintang5
-----
Hotel JINGGA
Kota Saranjana
Harga Rp.500000
Rating Bintang4
-----
Hotel BIRU
Kota Saranjana
Harga Rp.1000000
Rating Bintang3
-----
```