

**LAPORAN PRAKTIKUM**  
**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**  
**PERTEMUAN 6 : SORTING (BUBBLE, SELECTION, INSERTION)**



**KAYLA RACHMAUDINA SATITI PUTRI**

**2341760103**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI POLITEKNIK**  
**NEGERI MALANG**

**2024**



## JOB SHEET - 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

```

1  public class Mahasiswa15 {
2      String nama;
3      int thnMasuk, umur;
4      double ipk;
5
6      Mahasiswa15(String n, int t, int u, double i) {
7          nama = n;
8          thnMasuk = t;
9          umur = u;
10         ipk = i;
11     }
12
13     void tampil() {
14         System.out.println("Nama      : " + nama);
15         System.out.println("Tahun masuk : " + thnMasuk);
16         System.out.println("Umur      : " + umur);
17         System.out.println("IPK       : " + ipk);
18     }
19 }

```

4. Buat class DaftarMahasiswaBerprestasi

```

1  public class DaftarMahasiswaBerprestasi15 {
2      Mahasiswa15 listMhs[] = new Mahasiswa15[5];
3      int idx;
4

```

5. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.

```

5      void tambah(Mahasiswa15 m) {
6          if (idx < listMhs.length) {
7              listMhs[idx] = m;
8              idx++;
9          } else {
10             System.out.println("Data sudah penuh!!");
11         }
12     }

```

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.

```

14     void tampil() {
15         for (Mahasiswa15 m : listMhs) {
16             m.tampil();
17             System.out.println("-----");
18         }
19     }

```

7. Tambahkan method bubbleSort() di dalam class tersebut!

```

21     void bubbleSort() {
22         for (int i = 0; i < listMhs.length - 1; i++) {
23             for (int j = 1; j < listMhs.length - i; j++) {
24                 if (listMhs[j].ipk > listMhs[j-1].ipk) {
25                     // proses swap
26                     Mahasiswa15 tmp = listMhs[j];
27                     listMhs[j] = listMhs[j-1];
28                     listMhs[j-1] = tmp;
29                 }
30             }
31         }
32     }

```

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

```

1  public class Main15 {
2      public static void main(String[] args) {

```



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.

```

3      DaftarMahasiswaBerprestasi15 list = new DaftarMahasiswaBerprestasi15();
4      Mahasiswa15 m1 = new Mahasiswa15("Nusa", 2017, 25, 3);
5      Mahasiswa15 m2 = new Mahasiswa15("Rara", 2012, 19, 4);
6      Mahasiswa15 m3 = new Mahasiswa15("Dompur", 2018, 19, 3.5);
7      Mahasiswa15 m4 = new Mahasiswa15("Abdul", 2017, 23, 2);
8      Mahasiswa15 m5 = new Mahasiswa15("Ummi", 2019, 21, 3.75);
9
10     list.tambah(m1);
11     list.tambah(m2);
12     list.tambah(m3);
13     list.tambah(m4);
14     list.tambah(m5);
15
16     System.out.println("\nData mahasiswa sebelum sorting : \n");
17     list.tampil();
18
19     System.out.println("\nData mahasiswa setelah bubble sorting desc berdasarkan ipk : \n");
20     list.bubbleSort();
21     list.tampil();
22 }
23 }
    
```

### 5.2.2 Verifikasi Hasil 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      : 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 bubble 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
-----
```

### 5.2.3 Pertanyaan

1. Terdapat di method apakah proses bubble sort?

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?

Mengurutkan data dari array listMhs berdasarkan ipk, diurutkan dari yang terkecil hingga terbesar.

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?

Looping i merupakan outer loop yang berfungsi untuk mengatur iterasi perulangan secara keseluruhan

Looping j merupakan inner loop yang berfungsi untuk membandingkan elemen-elemen yang berdekatan.

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

Mencegah infinite loop dan memastikan elemen terakhir tidak terbandingkan

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

Mencegah infinite loop dan menyesuaikan perbandingan dengan elemen yang valid

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

Looping i akan berlangsung selama 49 kali dan jumlah tahap bubble sort yang ditempuh juga 49 kali, karena terdapat 49 elemen (dari 0 hingga 48) yang perlu dibandingkan, maka terdapat 49 tahap bubble sort.

### 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.

```

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         }
9         // swap
10        Mahasiswa15 tmp = listMhs[idxMin];
11        listMhs[idxMin] = listMhs[i];
12        listMhs[i] = tmp;
13    }
14 }
```



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

```
1 System.out.println("\nData mahasiswa setelah selection sorting asc berdasarkan ipk : \n");
2 list.selectionSort();
3 list.tampil();
```

- Coba jalankan kembali class Main, dan amati hasilnya! Apakah kini data mahasiswa telah tampilurut menaik berdasar ipk?

Iya, berikut ini adalah outputnya

### 5.3.2. Verifikasi Hasil 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 : 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 selection 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
-----
```

### 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!

Kode program tersebut melakukan proses untuk menemukan indeks elemen dengan IPK terkecil dalam listMhs dimulai dari indeks i.

Pertama, variabel idxMin diinisialisasi dengan nilai i (diasumsikan sebagai elemen terkecil). Kemudian, loop for dijalankan dari i + 1 hingga listMhs.length - 1 untuk mengabaikan elemen i dan memeriksa elemen selanjutnya.

Di dalam looping, IPK elemen dengan indeks j dibandingkan dengan IPK elemen idxMin. Jika IPK elemen j lebih kecil, maka nilai idxMin diperbarui dengan nilai j.

Pada akhir looping, idxMin akan berisi indeks elemen dengan IPK terkecil.

## 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.

```

1 void insertionSort(){
2     for (int i = 1; i < listMhs.length; i++) {
3         Mahasiswa15 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 }
```

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

```

1 System.out.println("\nData mahasiswa setelah insertion sorting asc berdasarkan ipk : \n");
2 list.insertionSort();
3 list.tampil();
```

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

Iya betul, berikut ini adalah outputnya



#### 5.4.2 Verifikasi Hasil 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      : 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 insertion 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.

```

1 void insertionSort(boolean asc){
2     for (int i = 1; i < listMhs.length; i++) {
3         Mahasiswa15 temp = listMhs[i];
4         int j = i;
5         if (asc) {
6             while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
7                 listMhs[j] = listMhs[j - 1];
8                 j--;
9             }
10        } else {
11            while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
12                listMhs[j] = listMhs[j - 1];
13                j--;
14            }
15        }
16        listMhs[j] = temp;
17    }
18 }

```

```

1 System.out.println("\nData mahasiswa setelah insertion sorting desc berdasarkan ipk : \n");
2 list.insertionSort(false);
3 list.tampil();

```

Data mahasiswa setelah insertion 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      : Dompur
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**.

