

Laporan Praktikum Algoritma & Struktur Data

Jobsheet 6



Nama: Aqil Rahmat Alifiandi

NIM: 2341760099

Prodi: D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023**

- **Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort**

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

```
package minggu5;

public class Mahasiswa {
    String nama;
    int thnMasuk, umur;
    double ipk;

    Mahasiswa(String n, int t, int u, double i){
        nama = n;
        thnMasuk = t;
        umur = u;
        ipk = i;
    }

    void tampil(){
        System.out.println("Nama = "+nama);
        System.out.println("Tahun Masuk = "+thnMasuk);
        System.out.println("Umur = "+umur);
        System.out.println("IPK = "+ipk);
    }
}
```

4. Buat class DaftarMahasiswaBerprestasi seperti di bawah ini!

```
package minggu5;

public class DaftarMahasiswaBerprestasi {
    Mahasiswa listMhs[] = new Mahasiswa[5];
    int idx;

    //setelah ini tuliskan method tambah()

    //setelah ini tuliskan method tampil()

    //setelah ini tuliskan method bubbleSort()
}
```

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

```
//setelah ini tuliskan method tambah()
void tambah(Mahasiswa m){
    if(idx<listMhs.length){
        listMhs[idx] = m;
        idx++;
    }else{
        System.out.println("Data sudah penuh!!");
    }
}
```

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.

```
//setelah ini tuliskan method tampil()
void tampil(){
    for(Mahasiswa m : listMhs){
        m.tampil();
        System.out.println("-----");
    }
}
```

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

```
//setelah ini tuliskan method bubbleSort()
void bubbleSort(){
    for(int i=0; i<listMhs.length-1; i++){
        for(int j=1; j<listMhs.length-i; j++){
            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;
            }
        }
    }
}
```

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

```
package minggu5;
import java.util.Scanner;

public class Main {
    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.

```

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();

```

Hasil coding & run:

Mahasiswa05:

```

public class Mahasiswa05 {

    String nama;
    int thnMasuk;
    int umur;
    double ipk;

    Mahasiswa05(String n, int t, int u, double i){
        nama = n;
        thnMasuk = t;
        umur = u;
        ipk = i;
    }

    void tampil(){
        System.out.println("Nama = "+nama);
        System.out.println("Tahun Masuk = "+thnMasuk);
        System.out.println("Umur = "+umur);
        System.out.println("IPK = "+ipk);
    }
}

```

DaftarMahasiswaBerprestasi05:

```

public class DaftarMahasiswaBerprestasi05 {
    Mahasiswa05 listMhs[] = new Mahasiswa05[5];
    int idx;
    void tambah(Mahasiswa05 m){
        if(idx<listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else{
            System.out.println(x:"Data Sudah Penuh!!");
        }
    }
    void tampil(){
        for(Mahasiswa05 m : listMhs){
            m.tampil();
            System.out.println(x:"-----");
        }
    }
    void bubbleSort(){
        for(int i=0; i<listMhs.length-1; i++){
            for(int j=1; j < listMhs.length-i; j++){
                if(listMhs[j].ipk < listMhs[j-1].ipk){
                    Mahasiswa05 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}

```

MainMahasiswa05:

```

import java.util.Scanner;

public class MainMahasiswa05 {
    Run | Debug
    public static void main(String[] args) {
        DaftarMahasiswaBerprestasi05 list = new DaftarMahasiswaBerprestasi05();
        Mahasiswa05 m1 = new Mahasiswa05(n:"Nusa", t:2017, u:25, i:3);
        Mahasiswa05 m2 = new Mahasiswa05(n:"Rara", t:2012, u:19, i:4);
        Mahasiswa05 m3 = new Mahasiswa05(n:"Dompu", t:2018, u:19, i:3.5);
        Mahasiswa05 m4 = new Mahasiswa05(n:"Abdul", t:2017, u:23, i:2);
        Mahasiswa05 m5 = new Mahasiswa05(n:"Ummi", t:2019, u:21, i:3.75);

        list.tambah(m1);
        list.tambah(m2);
        list.tambah(m3);
        list.tambah(m4);
        list.tambah(m5);

        System.out.println(x:"Data Mahasiswa Sebelum Sorting = ");
        list.tampil();

        System.out.println(x:"Data Mahasiswa Setelah Sorting desc berdasarkan IPK");
        list.bubbleSort();
        list.tampil();
    }
}

```

Run:

```

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 = Dompus
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?

Jawaban: terdapat di method bubblesort();

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?

Jawaban: untuk menukar nilai/swap nilai semua dengan hasil penyortiran

3. Perhatikan perulangan di dalam bubbleSort() di bawah ini:

```

for(int i=0; i<listMhs.length-1; i++){
    for(int j=1; j<listMhs.length-i; j++){

```

- a. Apakah perbedaan antara kegunaan perulangan i dan perulangan j?

Jawaban: Perulangan i digunakan untuk menunjukkan tahapan. Sedangkan pada perulangan j menunjukkan Langkah dalam setiap tahapan.

- b. Mengapa syarat dari perulangan i adalah i<listMhs.length-1?

Jawaban: Karena, perulangan i menunjukkan tahapan sehingga tahapan yang akan dilalui adalah "banyak_data - 1".

- c. Mengapa syarat dari perulangan j adalah j<listMhs.length-i?

Jawaban: Karena, perulangan j menunjukkan Langkah dalam setiap tahapan sehingga Langkah yang akan dilalui dalam setiap tahapan adalah 'banyak_data tahapan - 1'

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

Jawaban: Perulangan i akan berlangsung selama 49 kali dengan total tahapan yang akan dilalui 49 tahapan.

• Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort

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.

```
void selectionSort(){
    for(int i=0; i<listMhs.length-1; i++){
        int idxMin = i;
        for(int j=i+1; j<listMhs.length; j++){
            if(listMhs[j].ipk < listMhs[idxMin].ipk){
                idxMin = j;
            }
        }
        Mahasiswa05 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

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

```
System.out.println(x:"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?

<p>Data Mahasiswa Sebelum Sorting =</p> <p>Nama = Nusa Tahun Masuk = 2017 Umur = 25 IPK = 3.0</p> <p>-----</p> <p>Nama = Rara Tahun Masuk = 2012 Umur = 19 IPK = 4.0</p> <p>-----</p> <p>Nama = Dompus Tahun Masuk = 2018 Umur = 19 IPK = 3.5</p> <p>-----</p> <p>Nama = Abdul Tahun Masuk = 2017 Umur = 23 IPK = 2.0</p> <p>-----</p> <p>Nama = Ummi Tahun Masuk = 2019 Umur = 21 IPK = 3.75</p> <p>-----</p>	<p>Data Mahasiswa Setelah Sorting asc berdasarkan IPK</p> <p>Nama = Abdul Tahun Masuk = 2017 Umur = 23 IPK = 2.0</p> <p>-----</p> <p>Nama = Nusa Tahun Masuk = 2017 Umur = 25 IPK = 3.0</p> <p>-----</p> <p>Nama = Dompus Tahun Masuk = 2018 Umur = 19 IPK = 3.5</p> <p>-----</p> <p>Nama = Ummi Tahun Masuk = 2019 Umur = 21 IPK = 3.75</p> <p>-----</p> <p>Nama = Rara Tahun Masuk = 2012 Umur = 19 IPK = 4.0</p> <p>-----</p>
--	--

Pertanyaan

Di dalam method selection sort, terdapat baris program seperti di bawah ini:

```
int idxMin = i;
for(int j=i+1; j<listMhs.length; j++){
    if(listMhs[j].ipk < listMhs[idxMin].ipk){
        idxMin = j;
    }
}
```

Untuk apakah proses tersebut, jelaskan!

Jawaban: untuk mencari indeks dari elemen dengan nilai terkecil dalam array listMhs mulai dari indeks '1'.

• Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

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.


```

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa05 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}

```

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

```

System.out.println(x:"Data Mahasiswa Setelah Sorting asc berdasarkan IPK");
list.selectionSort();
list.tampil();

```

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

```

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

```

Pertanyaan

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

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

```
public class Hotel05 {  
    String nama, kota;  
    int harga;  
    byte bintang;  
  
    Hotel05(String n, String k, int h, Byte b){  
        nama = n;  
        kota = k;  
        harga = h;  
        bintang = b;  
    }  
  
    void tampil(){  
        System.out.println("Nama : " + nama);  
        System.out.println("Kota : " + kota);  
        System.out.println("Harga : " + harga);  
        System.out.println("Bintang : "+ bintang);  
    }  
}
```

```

public class HotelService05 {
    Hotel05 listHotel[] = new Hotel05[3];
    int idx;

    void tambah(Hotel05 z) {
        if (idx < listHotel.length) {
            listHotel[idx] = z;
            idx++;
        } else {
            System.out.println(x:"Data sudah penuh!!");
        }
    }

    void tampil() {
        for (Hotel05 z : listHotel) {
            z.tampil();
            System.out.println(x:"-----");
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listHotel.length - 1; i++) {
            for (int j = 1; j < listHotel.length - i; j++) {
                if (listHotel[j].bintang > listHotel[j - 1].bintang) {
                    Hotel05 tmp = listHotel[j];
                    listHotel[j] = listHotel[j - 1];
                    listHotel[j - 1] = tmp;
                }
            }
        }
    }

    void bubbleSort2() {
        for (int i = 0; i < listHotel.length - 1; i++) {
            for (int j = 1; j < listHotel.length - i; j++) {
                if (listHotel[j].harga > listHotel[j - 1].harga) {
                    Hotel05 tmp = listHotel[j];
                    listHotel[j] = listHotel[j - 1];
                    listHotel[j - 1] = tmp;
                }
            }
        }
    }

    void selectionSort() {
        for (int i = 0; i < listHotel.length - 1; i++) {
            int idxMin = i;
            for (int j = i + 1; j < listHotel.length; j++) {
                if (listHotel[j].bintang < listHotel[idxMin].bintang) {
                    idxMin = j;
                }
            }
            Hotel05 tmp = listHotel[idxMin];
            listHotel[idxMin] = listHotel[i];
            listHotel[i] = tmp;
        }
    }

    void selectionSort2() {
        for(int i = 0; i < listHotel.length-1; i++){
            int idxMin = i;
            for(int j=i+1; j < listHotel.length; j++){
                if(listHotel[j].harga < listHotel[idxMin].harga){
                    idxMin = j;
                }
            }

            Hotel05 tmp = listHotel[idxMin];
            listHotel[idxMin] = listHotel[i];
            listHotel[i] = tmp;
        }
    }
}

```

```

import java.util.Scanner;

public class MainHotel05 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc05 = new Scanner(System.in);

        HotelService05 list = new HotelService05();

        Hotel05 h1 = new Hotel05(n:"Ijen Suites", k:"Malang", h:1500000, (byte) 10);
        Hotel05 h2 = new Hotel05(n:"Grand Mercure", k:"Malang", h:2000000, (byte) 9);
        Hotel05 h3 = new Hotel05(n:"Golden Tulip", k:"Batu", h:1300000, (byte) 8);

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

        System.out.println(x:"Data hotel sebelum sorting:");
        list.tampil();

        System.out.println(x:"=====");

        System.out.println(x:"Data hotel setelah sorting desc berdasarkan bintang:");
        list.bubbleSort();
        list.tampil();

        System.out.println(x:"Data hotel setelah sorting desc berdasarkan harga:");
        list.bubbleSort2();
        list.tampil();

        System.out.println(x:"=====");

        System.out.println(x:"Data hotel setelah sorting desc berdasarkan bintang:");
        list.selectionSort();
        list.tampil();

        System.out.println(x:"Data hotel setelah sorting desc berdasarkan harga:");
        list.selectionSort2();
        list.tampil();

        sc05.close();
    }
}

```

Hasil Run:

Data hotel sebelum sorting:

Nama : Ijen Suites

Kota : Malang

Harga : 1500000

Bintang : 10

Nama : Grand Mercure

Kota : Malang

Harga : 2000000

Bintang : 9

Nama : Golden Tulip

Kota : Batu

Harga : 1300000

Bintang : 8

=====

Data hotel setelah sorting desc berdasarkan bintang:

Nama : Ijen Suites

Kota : Malang

Harga : 1500000

Bintang : 10

Nama : Grand Mercure

Kota : Malang

Harga : 2000000

Bintang : 9

Nama : Golden Tulip

Kota : Batu

Harga : 1300000

Bintang : 8

Data hotel setelah sorting desc berdasarkan harga:

Nama : Grand Mercure

Kota : Malang

Harga : 2000000

Bintang : 9

Nama : Ijen Suites

Kota : Malang

Harga : 1500000

Bintang : 10

Nama : Golden Tulip

Kota : Batu

Harga : 1300000

Bintang : 8

=====

Data hotel setelah sorting desc berdasarkan bintang:

Nama : Golden Tulip

Kota : Batu

Harga : 1300000

Bintang : 8

Nama : Grand Mercure

Kota : Malang

Harga : 2000000

Bintang : 9

Nama : Ijen Suites

Kota : Malang

Harga : 1500000

Bintang : 10

Data hotel setelah sorting desc berdasarkan harga:

Nama : Golden Tulip

Kota : Batu

Harga : 1300000

Bintang : 8

Nama : Ijen Suites

Kota : Malang

Harga : 1500000

Bintang : 10

Nama : Grand Mercure

Kota : Malang

Harga : 2000000

Bintang : 9
