

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET PERTEMUAN KE-5



RIO TRI PRAYOGO

TI 1A

26

2341720236

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

Sorting (Bubble, Selection, dan Insertion Sort)

Praktikum 1 : Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort

Percobaan :

Main

```
package minggu6;

public class Main26 {

    public static void main(String[] args) {

        DaftarMahasiswaBerprestasi list = new DaftarMahasiswaBerprestasi();

        Mahasiswa m1 = new Mahasiswa("Nusa", 2017, 25, 3);
        Mahasiswa m2 = new Mahasiswa("Rara", 2012, 19, 4);
        Mahasiswa m3 = new Mahasiswa("Dompu", 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 (Bubble Sort): ");
        list.bubbleSort();
        list.tampil();
    }
}
```

Mahasiswa

```
package minggu6;

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);
    }
}

```

DaftarMahasiswaBerprestasi

```

package minggu6;

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

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

    void tampil() {
        for (Mahasiswa m : listMhs) {
            m.tampil();
            System.out.println("-----");
        }
    }
}

```

```

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) {
                Mahasiswa tmp = listMhs[j];
                listMhs[j] = listMhs[j - 1];
                listMhs[j - 1] = tmp;
            }
        }
    }
}
}

```

Output :

Data mahasiswa sebelum sorting:	Data mahasiswa setelah sorting desc berdasarkan ipk (Bubble Sort):
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: Dompus Tahun masuk: 2018 Umur: 19 IPK: 3.5	Nama: Dompus 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
-----	-----

Pertanyaan :

1. Terdapat di method apakah proses bubble sort?
 - Proses bubble sort dalam kode diatas terjadi pada method `bubbleSort()` ; dan dipanggil dalam main pada baris `list.bubbleSort()` ;
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 |     }

```

- Baris diatas berfungsi untuk membandingkan (*compare*) dan menukar (*swap*). Baris *if/else* melakukan perbandingan/pengecekan apakah ipk yang sekarang lebih besar daripada ipk sebelumnya, jika iya maka masuk ke tahap selanjutnya. Pada tahap selanjutnya adalah melakukan penukaran. Baris pertama berfungsi untuk menyalin nilai elemen pada `listMhs[j]` ke dalam variabel sementara `tmp`. Lalu pada tahap selanjutnya melakukan pemindahan nilai elemen `listMhs[j-1]` kedalam `listMhs[j]`. Setelah itu nilai sebelumnya yang telah disimpan didalam `tmp` dipindahkan ke dalam variabel `listMhs[j-1]`. Dengan proses tersebut maka sudah terjadi penukaran (*swap*).

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`?
 - Kedua perulangan terletak pada jenis prosesnya. Perulangan '`j`' merupakan *inner loop* dimana melakukan proses membandingkan (*compare*) dan menukar (*swap*), proses ini hanya dilakukan satu kali dan hanya membandingkan dua variabel saja. Perulangan '`i`' merupakan *outer loop* dimana setelah melakukan perulangan '`j`' maka akan kembali ke perulangan '`i`' untuk melakukan proses pada perulangan '`j`' pada index selanjutnya.
- Mengapa syarat dari perulangan `i` adalah `i<listMhs.length-1` ?
 - Perulangan diatas akan berhenti saat `i<listMhs.length-1` hal ini berarti perulangan akan berhenti sebelum mencapai index akhir. Hal tersebut disebabkan pada saat perulangan mencapai index akhir maka otomatis index akhir tersebut sudah posisi pengurutan yang benar sehingga tidak perlu melakukan iterasi lagi.
- Mengapa syarat dari perulangan `j` adalah `j<listMhs.length-i` ?
 - Perulangan diatas akan berhenti saat `j<listMhs.length-i`. Dikarenakan pada pada iterasi '`i`' sudah memastikan bahwa elemen terbesar sudah terurut pada posisi terakhir, maka elemen terakhir sudah tidak perlu dilakukan iterasi lagi.
- Jika banyak data di dalam `listMhs` adalah 50, maka berapakah perulangan `i` akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
 - Karena persyaratan perulangan '`i`' merupakan `i<listMhs.length-1` maka jika banyak data adalah 50, perulangan akan terjadi sebanyak 49 kali (50-1). Tahap *bubble sort* yang ditempuh juga sebanyak 49 kali. Hal ini dikarenakan setiap iterasi pada perulangan '`i`' merupakan satu tahap *bubble sort*, sehingga tahap *bubble sort* akan sebanding/sama dengan banyak perulangan yang terjadi pada perulangan '`i`'.

Praktikum 2 : Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort

Percobaan :

"Baris kode di bawah merupakan tambahan yang dimasukkan ke dalam kode Praktikum 1"

Main

```
...
        System.out.println("Data mahasiswa setelah sorting asc berdasarkan
ipk (Selection Sort): ");
        list.selectionSort();
        list.tampil();
...
```

DataMahasiswaBerprestasi

```
...  
  
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;  
            }  
        }  
        Mahasiswa tmp = listMhs[idxMin];  
        listMhs[idxMin] = listMhs[i];  
        listMhs[i] = tmp;  
    }  
}  
  
...
```

Output :

Data mahasiswa sebelum sorting:	Data mahasiswa setelah sorting asc berdasarkan ipk (Selection Sort):
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: Dompur Tahun masuk: 2018 Umur: 19 IPK: 3.5 -----	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 -----
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!

- Baris diatas berfungsi untuk melakukan pencarian nilai terkecil/minimum yang terdapat pada array. Baris pertama adalah melakukan deklarasi index minimum berupa `idxMin`. Setelah itu pada baris selanjutnya melakukan perulangan `for` yang dimulai dari `j=i+1` sampai index akhir dalam array `listMhs`. Setiap iterasi dalam perulangan tersebut akan dilakukan pengecekan melalui `if` pada baris selanjutnya. Pengecekan tersebut mengecek apakah nilai `ipk` dari index '`j`' lebih kecil daripada nilai `ipk` dari index `idxMin`. Jika iya maka nilai `idxMin` akan diperbarui menjadi nilai '`j`'.

Praktikum 3 : Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

Percobaan :

"Baris kode di bawah merupakan tambahan yang dimasukkan ke dalam kode Praktikum 1"

Main

```
...
        System.out.println("Data mahasiswa setelah sorting asc berdasarkan
ipk (Insertion Sort): ");
        list.insertionSort();
        list.tampil();
...
```

DataMahasiswaBerprestasi

```
...
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;
    }
}
...
```

Output :

Data mahasiswa sebelum sorting:	Data mahasiswa setelah sorting berdasarkan ipk (Insertion Sort):
Nama: Nusa	Nama: Abdul
Tahun masuk: 2017	Tahun masuk: 2017
Umur: 25	Umur: 23
IPK: 3.0	IPK: 2.0
-----	-----
Nama: Rara	Nama: Nusa
Tahun masuk: 2012	Tahun masuk: 2017
Umur: 19	Umur: 25
IPK: 4.0	IPK: 3.0
-----	-----
Nama: Dompu	Nama: Dompu
Tahun masuk: 2018	Tahun masuk: 2018
Umur: 19	Umur: 19
IPK: 3.5	IPK: 3.5
-----	-----
Nama: Abdul	Nama: Ummi
Tahun masuk: 2017	Tahun masuk: 2019
Umur: 23	Umur: 21
IPK: 2.0	IPK: 3.75
-----	-----
Nama: Ummi	Nama: Rara
Tahun masuk: 2019	Tahun masuk: 2012
Umur: 21	Umur: 19
IPK: 3.75	IPK: 4.0
-----	-----

Pertanyaan :

1. Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.
 - Untuk mengubah proses sorting pada InsertionSort diatas menjadi descending dengan cara mengubah syarat pada baris *while* yang sebelumnya lebih besar (>) menjadi lebih kecil (<) atau menjadi seperti kode dibawah:

```
...  
        while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {  
...
```

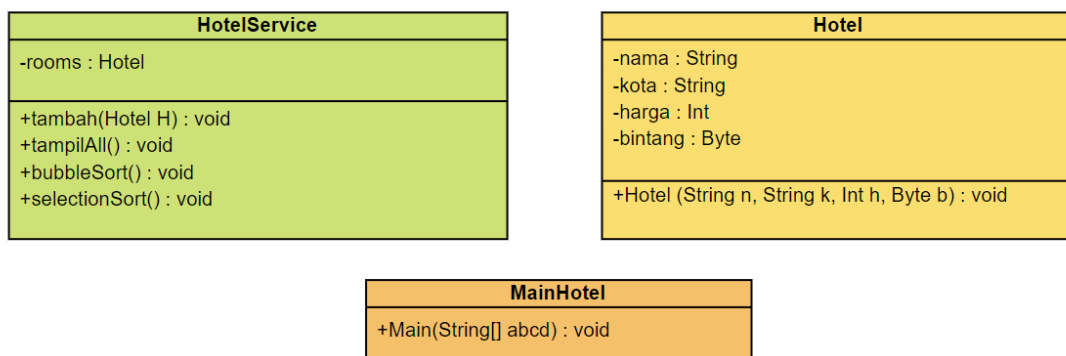
```
Data mahasiswa setelah sorting berdasarkan ipk (Insertion Sort):  
Nama: Rara  
Tahun masuk: 2012  
Umur: 19  
IPK: 4.0  
-----  
Nama: Ummi  
Tahun masuk: 2019  
Umur: 21  
IPK: 3.75  
-----  
Nama: Dompu  
Tahun masuk: 2018  
Umur: 19  
IPK: 3.5  
-----  
Nama: Nusa  
Tahun masuk: 2017  
Umur: 25  
IPK: 3.0  
-----  
Nama: Abdul  
Tahun masuk: 2017  
Umur: 23  
IPK: 2.0  
-----
```


Latihan Praktikum

Sebuah 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**.



Kode :

Main

```
package minggu6;

import java.util.Scanner;

public class MainHotel26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        HotelService list = new HotelService();

        Hotel h1 = new Hotel("Hotel Cahaya", "Jakarta", 3000000, (byte) 4);
        Hotel h2 = new Hotel("Hotel Bulan", "Bali", 5000000, (byte) 5);
        Hotel h3 = new Hotel("Hotel Bintang", "Malang", 1000000, (byte) 3);
        Hotel h4 = new Hotel("Hotel Matahari", "Surabaya", 800000, (byte)
1);
        Hotel h5 = new Hotel("Hotel Angkasa", "Bandung", 1500000, (byte)
2);
    }
}
```

```

list.tambah(h1);
list.tambah(h2);
list.tambah(h3);
list.tambah(h4);
list.tambah(h5);
boolean run = true;

do {
    System.out.println("\n-----");
    System.out.println("Selamat datang di program Pemesanan
Hotel");

    System.out.println("Pilih Hotel Berdasarkan:");
    System.out.println("1. Harga Termurah");
    System.out.println("2. Bintang Tertinggi");
    System.out.println("0. Keluar");
    System.out.print("(1/2/0): ");
    int input = scan.nextInt();
    switch (input) {
        case 1:
            System.out.println("-----");
            System.out.println("Pilihan hotel berdasarkan harga
termurah");

            list.bubbleSort();
            list.tampilAll();
            break;
        case 2:
            System.out.println("-----");
            System.out.println("Pilihan hotel berdasarkan bintang
tertinggi");

            list.selectionSort();
            list.tampilAll();
            break;
        case 0:
            System.out.println("-----");
            System.out.println("Terima Kasih!");
            run = false;
            break;
    }
}

```

```

        default:
            System.out.println("-----");
            System.out.println("Pilihan tidak valid");
            break;
    }
} while (run);
}
}

```

Hotel

```

package minggu6;

public class Hotel {
    String nama, kota;
    int harga;
    Byte bintang;

    Hotel(String n, String k, int h, Byte b) {
        nama = n;
        kota = k;
        harga = h;
        bintang = b;
    }

    void tampilAll() {
        System.out.println("Nama: " + nama);
        System.out.println("Kota: " + kota);
        System.out.println("Harga: " + harga);
        System.out.println("Bintang: " + bintang);
    }
}

```

HotelService

```

package minggu6;

public class HotelService {
    Hotel rooms[] = new Hotel[5];
    int idh;
}

```

```

void tambah(Hotel h) {
    if (idh < rooms.length) {
        rooms[idh] = h;
        idh++;
    } else {
        System.out.println("Data sudah penuh");
    }
}

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

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 idhMin = i;
        for (int j = i + 1; j < rooms.length; j++) {
            if (rooms[j].harga > rooms[idhMin].harga) {
                idhMin = j;
            }
        }
    }
}

```

```
        }

        Hotel temp = rooms[idhMin];

        rooms[idhMin] = rooms[i];

        rooms[i] = temp;

    }

}
```

Output :

Harga Termurah (Bubble Sort)	Bintang Tertinggi (Selection Sort)
Selamat datang di program Pemesanan Hotel ----- Pilih Hotel Berdasarkan: 1. Harga Termurah 2. Bintang Tertinggi 0. Keluar (1/2/0): 1 ----- Pilihan hotel berdasarkan harga termurah ----- Nama: Hotel Matahari Kota: Surabaya Harga: 800000 Bintang: 1 ----- Nama: Hotel Bintang Kota: Malang Harga: 1000000 Bintang: 3 ----- Nama: Hotel Angkasa Kota: Bandung Harga: 1500000 Bintang: 2 ----- Nama: Hotel Cahaya Kota: Jakarta Harga: 3000000 Bintang: 4 ----- Nama: Hotel Bulan Kota: Bali Harga: 5000000 Bintang: 5 -----	Selamat datang di program Pemesanan Hotel ----- Pilih Hotel Berdasarkan: 1. Harga Termurah 2. Bintang Tertinggi 0. Keluar (1/2/0): 2 ----- Pilihan hotel berdasarkan bintang tertinggi ----- Nama: Hotel Bulan Kota: Bali Harga: 5000000 Bintang: 5 ----- Nama: Hotel Cahaya Kota: Jakarta Harga: 3000000 Bintang: 4 ----- Nama: Hotel Angkasa Kota: Bandung Harga: 1500000 Bintang: 2 ----- Nama: Hotel Bintang Kota: Malang Harga: 1000000 Bintang: 3 ----- Nama: Hotel Matahari Kota: Surabaya Harga: 800000 Bintang: 1 -----