

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET 5**



**NAMA : DIMAS ADI BAYU SAMUDRA**

**KELAS : 1A**

**NO. ABSEN : 08**

**NIM : 2341720169**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## Percobaan 1

### Code mahasiswa

```
package jobsheet5.bubble;

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);
    }
}
```

### Code DaftarMahasiswa

```
package jobsheet5.bubble;

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;
            }
        }
    }
}
}
}

```

### Code mainMahasiswa

```

package jobsheet5.bubble;

public class mainMahasiswa {
    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 di sorting = ");
        list.tampil();
    }
}

```

```
        System.out.println("Data Mahasiswa setelah desc berdasarkan IPK =
");
        list.bubblesort();
        list.tampil();
    }
}
```

## Hasil run

Data Mahasiswa sebelum di 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 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. 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 |     }
```

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?
- 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. Terdapat pada method bubblesort () pada class daftarMahasiswaBerprestasi
2. Untuk mengurutkan data dari IPK yang terbesar sampai yang terkecil dengan mengecek apakah IPK data sebelumnya sudah lebih besar atau tidak, jika iya maka data yang ingin di ganti akan di simpan di tmp dan data akan di ubah lalu data yang sudah disimpan pada temp akan di isi ke data yang dipindahkan tadi, namun jika tidak maka akan mengecek hingga data itu menemukan yang lebih kecil darinya.
- 3.a. Perulangan j digunakan untuk melakukan pengecekan dan melakukan pergantian hingga data ke 0 sudah menjadi data yang terbesar, sedangkan perulangan i digunakan jika data perulangan j sudah selesai dan data ke 0 sudah benar maka akan di ubah agar perulangan j mengecek dan mengurutkan data ke 1 hingga selesai.
- 3.b. Saat melakukan pengurutan maka listMhs.length harus dikurangi 1 dikarenakan data yang ingin di pindah tidak perlu dihitung , jadi misal ada 8 data maka pengurutan di lakukan sebanyak 7 kali.
- 3.c. Perulangan yang berlangsung sebanyak 49 kali dan tahap yang di tempuh sebanyak 49 kali jika tahap dimulai dari tahap 1, tetapi jika dimulai dari tahap 0 maka ada total tahap yang ditempuh sebanyak 48 kali.

## Percobaan 2

### Code tambahan pada daftarMahasiswaBerprestasi

```
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;
    }
}
```

### Code tambahan pada mainMahasiswa

```
System.out.println("Data Mahasiswa setelah sorting asc berdasarkan IPK");
    list.selectionSort();
    list.tampil();
```

### Hasil run

```
-----
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 : 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
-----
```

### 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!

### Jawaban

Jadi, kode ini yang digunakan untuk mencari elemen dengan nilai IPK terkecil di antara elemen-elemen yang belum diurutkan dalam array listMhs. Setelah elemen dengan nilai IPK terkecil ditemukan, kita menukar posisi elemen tersebut dengan elemen pada indeks ke-i, sehingga mengurutkan data IPK dari yang terkecil hingga terbesar.

### Percobaan 3

#### Code tambahan pada daftarMahasiswaBerprestasi

```
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;  
    }  
}
```

#### Code tambahan pada mainMahasiswa

```
System.out.println("Data Mahasiswa setelah sorting asc berdasarkan ipk");  
list.insertionSort();  
list.tampil();
```

## Hasil run

```
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

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

## Jawaban

### Code yang diubah

```
void insertionSortDesending(){
    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;
    }
}
```



## Hasil run

```
Data Mahasiswa setelah sorting decs 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
-----
```

## Latihan Praktikum

### Code Hotel

```
package jobsheet5.latihan;

public class Hotel {

    String nama;
    String kota;
    int harga;
    byte bintang;

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

    void tampil(){
        System.out.println("Nama Hotel = " +nama);
        System.out.println("Kota = " + kota);
        System.out.println("Harga sewa Hotel = " + harga);
    }
}
```

```
        System.out.println("Bintang Hotel = " + (byte)bintang);
    }
}
```

#### Code HotelService

```
package jobsheet5.latihan;

public class HotelService {

    Hotel listHotels[] = new Hotel[3];
    int rooms;

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

    void tampilAll(){
        for (Hotel H : listHotels) {
            H.tampil();
            System.out.println("-----");
        }
    }

    void bubblesort(){
        for (int i = 0; i < listHotels.length- 1; i++) {
            for (int j = 1; j < listHotels.length-i; j++) {
                if (listHotels[j].harga < listHotels[j-1].harga) {
                    Hotel tmp  = listHotels[j];
                    listHotels[j] = listHotels[j-1];
                    listHotels[j-1] = tmp;
                }
            }
        }
    }

    void selectionSort(){
        for (int i = 0; i < listHotels.length - 1; i++) {
```

```

        int roomsMin = i;
        for (int j = i + 1; j < listHotels.length; j++) {
            if (listHotels[j].bintang > listHotels[roomsMin].bintang) {
                roomsMin = j;
            }
        }
        Hotel tmp = listHotels[roomsMin];
        listHotels[roomsMin] = listHotels[i];
        listHotels[i] = tmp;
    }
}
}

```

#### Code MainHotel

```

package jobsheet5.latihan;
import java.util.Scanner;
public class MainHotel {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        HotelService list = new HotelService();
        Hotel H1 = new Hotel("Sibuya", "Jepang", 350000, (byte)5);
        Hotel H2 = new Hotel("Savana" , "Malang", 600000, (byte)4);
        Hotel H3 = new Hotel("Ariya" , "Malang", 500000, (byte)3);

        list.tambah(H1);
        list.tambah(H2);
        list.tambah(H3);

        list.tampilAll();
        System.out.println("Menu Filter Hotel ");
        System.out.println("1. Harga ");
        System.out.println("2. Bintang ");
        System.out.print("Masukan Pilihan anda : " );
        int pil = sc.nextInt();
        System.out.println("-----");
        System.out.println("");

        if (pil == 1) {

```

```

        list.bubblesort();
        list.tampilAll();
    } else if (pil == 2) {
        list.selectionSort();
        list.tampilAll();
    } else {
        System.out.println("Filter yang anda pilih tidak ada");
    }
}
}

```

Hasil run 1

```

Nama Hotel = Sibuya
Kota = Jepang
Harga sewa Hotel = 350000
Bintang Hotel = 5
-----
Nama Hotel = Savana
Kota = Malang
Harga sewa Hotel = 600000
Bintang Hotel = 4
-----
Nama Hotel = Ariya
Kota = Malang
Harga sewa Hotel = 500000
Bintang Hotel = 3
-----
Menu Filter Hotel
1. Harga
2. Bintang
Masukan Pilihan anda : 1
-----
Nama Hotel = Sibuya
Kota = Jepang
Harga sewa Hotel = 350000
Bintang Hotel = 5
-----
Nama Hotel = Ariya
Kota = Malang
Harga sewa Hotel = 500000
Bintang Hotel = 3
-----
Nama Hotel = Savana
Kota = Malang
Harga sewa Hotel = 600000
Bintang Hotel = 4
-----

```

Hasil run 2

```
Nama Hotel = Sibuya
Kota = Jepang
Harga sewa Hotel = 350000
Bintang Hotel = 5
-----
Nama Hotel = Savana
Kota = Malang
Harga sewa Hotel = 600000
Bintang Hotel = 4
-----
Nama Hotel = Ariya
Kota = Malang
Harga sewa Hotel = 500000
Bintang Hotel = 3
-----
Menu Filter Hotel
1. Harga
2. Bintang
Masukan Pilihan anda : 2
-----

Nama Hotel = Sibuya
Kota = Jepang
Harga sewa Hotel = 350000
Bintang Hotel = 5
-----
Nama Hotel = Savana
Kota = Malang
Harga sewa Hotel = 600000
Bintang Hotel = 4
-----
Nama Hotel = Ariya
Kota = Malang
Harga sewa Hotel = 500000
Bintang Hotel = 3
-----
```