

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET PERTEMUAN KE-6



NAMA : ALVINO VALERIAN D.R

KELAS : 1A

NO. ABSEN : 05

NIM : 2341720027

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

5.2.3 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. Proses bubble sort terdapat di method bubbleSort() pada class DaftarMahasiswaBerprestasi.
2. digunakan untuk menukar dua elemen array yang berdekatan jika elemen di sebelah kiri lebih besar daripada elemen di sebelah kanan. Proses ini dilakukan untuk memastikan bahwa elemen array diurutkan berdasarkan nilai ipk secara descending.

3.

a. Perulangan i:

Digunakan untuk mengulang setiap elemen dalam array listMhs. Berulang sebanyak $\text{listMhs.length} - 1$ kali. Mengontrol looping luar dari bubble sort.

Perulangan j:

Digunakan untuk membandingkan elemen $\text{listMhs}[j]$ dengan elemen $\text{listMhs}[j-1]$. Berulang sebanyak $\text{listMhs.length} - i$ kali. Mengontrol looping dalam dari bubble sort.

b. Syarat $i < \text{listMhs.length} - 1$ digunakan untuk memastikan bahwa perulangan i tidak melebihi batas array listMhs.

c. Syarat $j < \text{listMhs.length} - i$ digunakan untuk memastikan bahwa perulangan j tidak melebihi batas array listMhs dan tidak membandingkan elemen yang sama.

d. Jumlah Perulangan i:

Jika banyak data di dalam listMhs adalah 50, maka perulangan i akan berlangsung sebanyak 49 kali.

Perhitungan: $\text{listMhs.length} - 1 = 50 - 1 = 49$.

Jumlah Tahap Bubble Sort:

Jumlah tahap bubble sort tidak dapat ditentukan secara pasti. Tahap bubble sort berhenti ketika tidak ada lagi elemen yang ditukar. Pada kasus lain bubble sort membutuhkan $n-1$ tahap, di mana n adalah jumlah elemen dalam array.

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!

JAWABAN:

1. merupakan bagian dari algoritma selection sort yang digunakan untuk menemukan elemen terkecil dalam array listMhs dan menempatkannya di awal array.

idxMin = 1: Baris ini menginisialisasi variabel idxMin dengan nilai 1. Variabel idxMin digunakan untuk menyimpan indeks elemen terkecil yang ditemukan.

for(int j=i+1; j<listMhs.length; j++): Perulangan ini digunakan untuk mencari elemen terkecil dalam array listMhs dimulai dari indeks i+1 hingga indeks terakhir array.

if(listMhs[j].ipk < listMhs[idxMin].ipk): Kondisi if ini digunakan untuk membandingkan nilai ipk dari elemen listMhs[j] dengan elemen listMhs[idxMin]. Jika nilai ipk dari elemen listMhs[j] lebih kecil daripada elemen listMhs[idxMin], maka nilai idxMin diubah menjadi j.

idxMin = j: Baris ini mengubah nilai idxMin menjadi j jika elemen listMhs[j] memiliki nilai ipk terkecil.

5.4.3 Pertanyaan

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

```
void insertionSortDesc() {  
    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;  
    }  
}
```

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 = 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

D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet6>

LATIHAN

```
package latihan;

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 tampil(){
        System.out.println("nama = "+nama);
        System.out.println("Kota = "+kota);
        System.out.println("Harga = "+harga);
        System.out.println("Bintang = "+bintang);
    }
}
```

```
package latihan;

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

    void tambah(Hotel h){
        if (rooms<listHotel.length){
            listHotel[rooms] = h;
            rooms++;
        }else{
            System.out.println("Hotel Sudah Penuh!!");
        }
    }

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

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

    void bintangSelectionShort(){
        for(int i=0;i<listHotel.length;i++){
            int indxMin = i;
```

```

        for(int j = i+1;j<listHotel.length;j++){
            if(listHotel[j].bintang >
listHotel[indxMin].bintang){
                indxMin = j;
            }
        }
        // swap
        Hotel tmp = listHotel[indxMin];
        listHotel[indxMin] = listHotel[i];
        listHotel[i] = tmp;
    }
}
}

```

```

package latihan;

public class MainHotel {
    public static void main(String[] args) {
        HotelService list = new HotelService();
        Hotel h1 = new Hotel("Solaris", "Malang", 520000, (byte)3);
        Hotel h2 = new Hotel("Harris", "Malang", 1000000, (byte)2);
        Hotel h3 = new Hotel("De Batu", "Batu", 360000, (byte)1);
        Hotel h4 = new Hotel("Grand Heaven", "Surabaya", 1500000,
(byte)5);
        Hotel h5 = new Hotel("Ibis", "Malang", 600000, (byte)4);

        list.tambah(h1);
        list.tambah(h2);
        list.tambah(h3);
        list.tambah(h4);
        list.tambah(h5);

        System.out.println("");
        System.out.println("Data Sebelum Sorting = ");
        list.tampil();
    }
}

```

```

        System.out.println("Data Harga Hotel Setelah Sorting Asc");
        list.hargaBubbleShort();
        list.tampil();

        System.out.println("Data Bintang Hotel Setelah Sorting
Desc");
        list.bintangSelectionShort();
        list.tampil();
    }
}

```

```

Data Sebelum Sorting =
nama = Solaris
Kota = Malang
Harga = 520000
Bintang = 3
-----
nama = Harris
Kota = Malang
Harga = 1000000
Bintang = 2
-----
nama = De Batu
Kota = Batu
Harga = 360000
Bintang = 1
-----
nama = Grand Heaven
Kota = Surabaya
Harga = 1500000
Bintang = 5
-----
nama = Ibis
Kota = Malang
Harga = 600000
Bintang = 4
-----

```


Data Harga Hotel Setelah Sorting Asc

nama = De Batu

Kota = Batu

Harga = 360000

Bintang = 1

nama = Solaris

Kota = Malang

Harga = 520000

Bintang = 3

nama = Ibis

Kota = Malang

Harga = 600000

Bintang = 4

nama = Harris

Kota = Malang

Harga = 1000000

Bintang = 2

nama = Grand Heaven

Kota = Surabaya

Harga = 1500000

Bintang = 5

Data Bintang Hotel Setelah Sorting Desc

nama = Grand Heaven

Kota = Surabaya

Harga = 1500000

Bintang = 5

nama = Ibis

Kota = Malang

Harga = 600000

Bintang = 4

nama = Solaris

Kota = Malang

Harga = 520000

Bintang = 3

nama = Harris

Kota = Malang

Harga = 1000000

Bintang = 2

nama = De Batu

Kota = Batu

Harga = 360000

Bintang = 1

D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet6>