LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET 6 ASD Sorting(bubble, selection, and insertion sort)

Dosen Pengajar : Triana Fatmawati, S.T., M.T.

Muhammad Afiq Firdaus

2341760189 / 21

SIB-1E

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

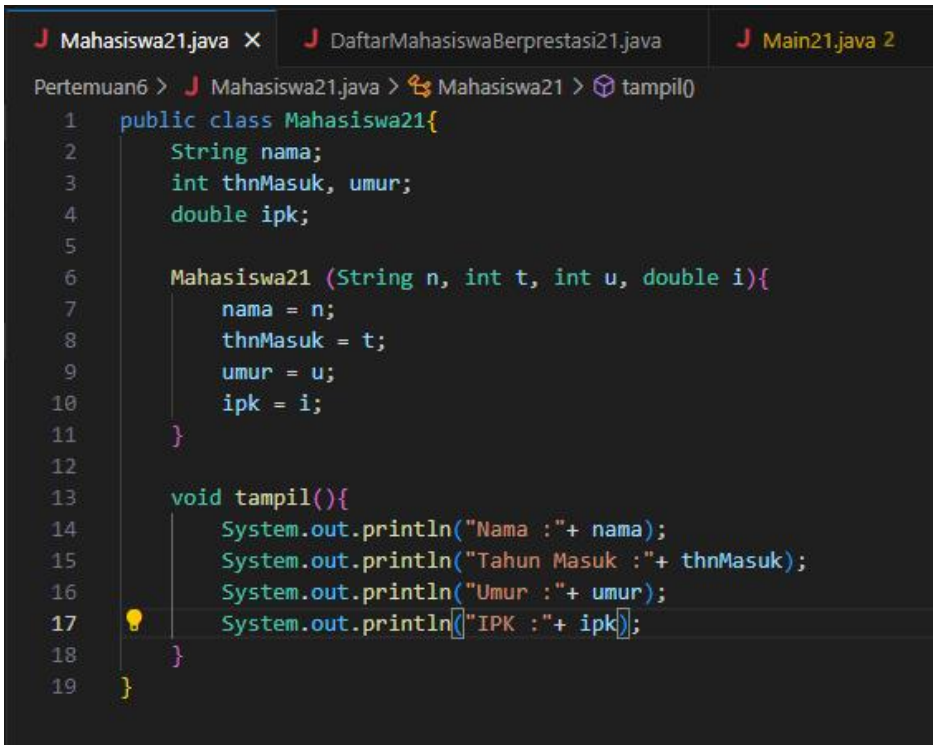
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

Percobaan 1 Bubble sort

Berikut adalah code Mahasiswa berdasarkan jobsheet



Berikut adalah code daftarMahasiswaBerprestasi berdasarkan jobsheet

Mahasiswa21.java | DaftarMahasiswaBerprestasi21.java | Main21.java 2

Pertemuan6 > DaftarMahasiswaBerprestasi21.java > DaftarMahasiswaBerprestasi21

```java
public class DaftarMahasiswaBerprestasi21 {
    Mahasiswa21 listMhs[] = new Mahasiswa21[5];
    int idx;

    void tambah(Mahasiswa21 m){
        if(idx<listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else{
            System.out.println(x:"Data sudah penuh!!");
        }
    }

    void tampil(){
        for(Mahasiswa21 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-1; j++){
                if(listMhs[j].ipk>listMhs[j-1].ipk){
                    Mahasiswa21 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}
```

Berikut adalah code Main berdasarkan jobsheet

Mahasiswa21.java | DaftarMahasiswaBerprestasi21.java | Main21.java 2

Pertemuan6 > Main21.java > Main21 > main(String[])

```java
import java.util.Scanner;

public class Main21 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        DaftarMahasiswaBerprestasi21 list = new DaftarMahasiswaBerprestasi21();
        Mahasiswa21 m1 = new Mahasiswa21(n:"Nusa", t:2017, u:25, i:3);
        Mahasiswa21 m2 = new Mahasiswa21(n:"Rara", t:2012, u:19, i:4);
        Mahasiswa21 m3 = new Mahasiswa21(n:"Dompu", t:2018, u:19, i:3.5);
        Mahasiswa21 m4 = new Mahasiswa21(n:"Abdul", t:2017, u:23, i:2);
        Mahasiswa21 m5 = new Mahasiswa21(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();
    }
}

```

Berikut adalah hasil output dari code diatas

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\
CodeDetailsInExceptionMessages' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\
\bin' 'Main21'
Data Mahasiswa sebelum sorting =
NamaNusa
Tahun Masuk2017
Umur =25
IPK =3.0
----------------------
NamaRara
Tahun Masuk2012
Umur =19
IPK =4.0
----------------------
NamaDompu
Tahun Masuk2018
Umur =19
IPK =3.5
----------------------
NamaAbdul
Tahun Masuk2017
Umur =23
IPK =2.0
----------------------
Tahun Masuk2019
Umur =21
IPK =3.75
----------------------
```

```
----------------------
Data Mahasiswa setelah sorting desc berdasarkan ipk
NamaRara
Tahun Masuk2012
Umur =19
IPK =4.0
----------------------
NamaDompu
Tahun Masuk2018
Umur =19
IPK =3.5
----------------------
NamaNusa
Tahun Masuk2017
Umur =25
IPK =3.0
----------------------
NamaAbdul
Tahun Masuk2017
Umur =23
IPK =2.0
----------------------
NamaUmmi
Tahun Masuk2019
Umur =21
IPK =3.75
----------------------
```

Pertanyaan :

1. Terdapat di method apakah proses bubble sort?
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?

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<listMhs.length-1?
c. Mengapa syarat dari perulangan j adalah j
d. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Jawaban :

1. Bubble sort berada di method bubble sort pada class DaftarMahasiswaBerprestasi

```
void bubblesort(){
    for(int i=0; i<listMhs.length-1; i++){
        for(int j=1; j<listMhs.length-1; j++){
            if(listMhs[j].ipk>listMhs[j-1].ipk){
                Mahasiswa21 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}
```

2. Proses tersebut untuk membandingkan dua elemen berturut-turut dalam array dan menukar posisinya jika elemen yang berada di indeks ke-j memiliki nilai IPK yang lebih tinggi daripada elemen yang berada di indeks ke-(j-1). Dengan cara ini, elemen dengan nilai IPK tertinggi akan "menggelembung" ke posisi yang benar secara berurutan, hingga seluruh array terurut.

3. a. Perbedaan antara variabel perulangan **i** dan **j** adalah bahwa **i** digunakan sebagai indeks untuk mengakses elemen-elemen array secara berurutan dari awal hingga akhir, sedangkan **j** digunakan sebagai indeks untuk membandingkan pasangan-pasangan elemen dalam algoritma bubble sort. Dalam implementasi ini, **i** digunakan untuk mengontrol perulangan luar yang mengatur jumlah iterasi keseluruhan, sementara **j** digunakan untuk mengontrol perulangan dalam yang digunakan untuk membandingkan elemen-elemen dan melakukan pertukaran.

   b. Syarat dari perulangan **i** adalah **i < listMhs.length-1** karena kita ingin memastikan bahwa perulangan tersebut berhenti sebelum mencapai elemen terakhir dalam array. Jika perulangan terus berlanjut hingga **i = listMhs.length**, itu berarti semua elemen sudah berada pada posisi yang benar dalam urutan terurut.

c. Syarat dari perulangan **j** adalah **j < listMhs.length-1** karena kita ingin memastikan bahwa kita hanya membandingkan elemen-elemen yang ada dalam array. Karena kita sudah membandingkan elemen-elemen pada posisi **j** dan **j-1**, maka kita tidak perlu membandingkan elemen pada posisi terakhir array (indeks **listMhs.length-1**), karena tidak ada elemen setelahnya.

d. Jika banyak data di dalam **listMhs** adalah 50, maka perulangan **i** akan berlangsung sebanyak 49 kali. Ini karena kita menggunakan kondisi **i < listMhs.length-1**, yang berarti perulangan akan terus berlanjut sampai **i** mencapai indeks terakhir sebelum akhir array. Tahap bubble sort yang ditempuh adalah sama dengan jumlah iterasi perulangan **i**, yaitu 49 tahap.

Percobaan 2 Selection Sort

Berikut adalah code yang ditambahkan pada percobaan 2

```
                    Mahasiswa21 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa21 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
}
```

Yaitu ada penambahan selectionsort tetapi code sebelumnya masih menggunakan code yang sama.

Dan berikut adalah penambahan code pada percobaan 2

```
    System.out.println(x:"Data Mahasiswa setelah sorting asc berdasarkan ipk");
    list.selectionSort();
    list.tampil();
}
```

Yaitu pada class main tetapi code sebelumnya masih menggunakan code yang sama yang digunakan pada percobaan sebelumnya.

Dan berikut adalah hasil output dari code tersebut, output yang ditampilkan adalah hanya tambahan dari code percobaan 2

```
-----------------------
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
-----------------------
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 6>
```

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 :

Proses tersebut untuk mencari elemen terkecil dalam sublist yang belum diurutkan pada setiap iterasi dari algoritma selection sort. Setelah iterasi ke-i dari perulangan luar (yang mengontrol posisi awal dari sublist yang belum diurutkan), kita asumsikan bahwa elemen ke-i adalah elemen terkecil yang kita temui. Kemudian, kita melakukan perulangan dalam (yang dimulai dari i+1) untuk mencari elemen terkecil dalam sublist yang belum diurutkan. Setiap kali kita menemukan elemen yang lebih kecil dari elemen ke-i yang sedang kita pertimbangkan, kita memperbarui indeks elemen terkecil (idxMin) menjadi indeks elemen tersebut. Setelah perulangan dalam selesai, kita menukar elemen ke-i dengan elemen terkecil yang ditemukan (elemen di indeks idxMin). Hal ini memastikan bahwa setelah iterasi ke-i dari perulangan luar, elemen ke-i akan menjadi elemen terkecil dalam sublist yang belum diurutkan. Proses ini diulang hingga seluruh array terurut.

Percobaan 3 Insertion Sort

Berikut adalah hasil code percobaan 3 berdasarkan jobsheet, code yang ditampilkan adalah code tambahan saja pada percobaan 3

```
        System.out.println("x: ----------------------");
    }
}

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa21 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
}
```

```
        Mahasiswa21 m5 = new Mahasiswa21(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 asc berdasarkan ipk");
        list.insertionSort();
        list.tampil();
    }
}
```

Berikut adalah hasil output dari code diatas

```
'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:-
607894c15e563992b66ea6\redhat.java\jdt_ws\Jobsheet 6_399f75b6\bin' 'Main2
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
----------------------
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\
```

Pertanyaan :

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

Jawaban :

```
            System.out.println(x:"----------------------");
        }
    }

    void insertionSortDescending() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa21 temp = listMhs[i];
            int j = i;
            while (j > 0 && listMhs[j - 1].ipk < temp.ipk) { // Mengu
                listMhs[j] = listMhs[j - 1];
                j--;
            }
            listMhs[j] = temp;
        }
    }

}
```

Berikut adalah hasil output dari code diatas

```
Data Mahasiswa setelah sorting asc 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
----------------------
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktik
```

5.5 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.**

Berikut adalah code dari HotelService, Hotel, Main.

Code HotelService

Pertemuan6 > HotelService.java > ...

```java
public class HotelService {
    private Hotel[] rooms;
    private int count;

    public HotelService(int size) {
        rooms = new Hotel[size];
        count = 0;
    }

    public void tambah(Hotel H) {
        if (count < rooms.length) {
            rooms[count] = H;
            count++;
        } else {
            System.out.println(x:"Kamar penuh, tidak dapat menambahkan lagi.");
        }
    }

    public void tampilAll() {
        for (int i = 0; i < count; i++) {
            System.out.println("Nama Hotel: " + rooms[i].nama + ", Kota: " + rooms[i].kota + ", Harga: "
                    + rooms[i].harga + ", Bintang: " + rooms[i].bintang);
        }
    }

    public void bubbleSort() {
        for (int i = 0; i < count - 1; i++) {
            for (int j = 0; j < count - 1 - i; j++) {
                if (rooms[j].harga > rooms[j + 1].harga) {
                    Hotel temp = rooms[j];
                    rooms[j] = rooms[j + 1];
                    rooms[j + 1] = temp;
                }
            }
        }
    }

    public void selectionSort() {
        for (int i = 0; i < count - 1; i++) {
            int idxMin = i;
            for (int j = i + 1; j < count; j++) {
                if (rooms[j].bintang > rooms[idxMin].bintang) {
                    idxMin = j;
                }
            }
            Hotel temp = rooms[i];
            rooms[i] = rooms[idxMin];
            rooms[idxMin] = temp;
        }
    }
}
```

Hotel[] rooms

Code Hotel

```java
public class Hotel {
    String nama;
    String kota;
    int harga;
    byte bintang;

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

Code Main

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        HotelService service = new HotelService(size:5);

        service.tambah(new Hotel(n:"Hotel A", k:"Jakarta", h:500000, (byte) 4));
        service.tambah(new Hotel(n:"Hotel B", k:"Surabaya", h:300000, (byte) 3));
        service.tambah(new Hotel(n:"Hotel C", k:"Bali", h:700000, (byte) 5));
        service.tambah(new Hotel(n:"Hotel D", k:"Yogyakarta", h:400000, (byte) 2));
        service.tambah(new Hotel(n:"Hotel E", k:"Bandung", h:600000, (byte) 4));

        System.out.println(x:"Sebelum sorting:");
        service.tampilAll();

        System.out.println(x:"\nSetelah sorting berdasarkan harga (bubble sort):");
        service.bubbleSort();
        service.tampilAll();

        System.out.println(x:"\nSetelah sorting berdasarkan bintang (selection sort):");
        service.selectionSort();
        service.tampilAll();
    }
}
```

Berikut adalah output dari Code tersebut