

Laporan Praktikum ASD Jobsheet 06

Nama: Gegas Anugrah Derajat

Kelas: SIB-1F

Percobaan 5.2

Membuat class, menambahkan atribut dan konstruktor

```
1 public class Mahasiswa11 {
2     String nama;
3     int thnMasuk, umur;
4     double ipk;
5     Mahasiswa11(String n, int t, int u, double i) {
6         nama = n;
7         thnMasuk = t;
8         umur = u;
9         ipk = i;
10    }
11
12    void tampil() {
13        System.out.println("Nama = "+nama);
14        System.out.println("Tahun masuk = "+thnMasuk);
15        System.out.println("Umur = "+umur);
16        System.out.println("IPK = "+ipk);
17    }
18 }
```

Membuat class DaftarMahasiswaBerprestasi

```
public class DaftarMahasiswaBerprestasi11 {
    Mahasiswa11 listMhs[] = new Mahasiswa11[5];
    int idx;
```

Menambahkan method tambah() dan method tampil()

```
    void tambah(Mahasiswa11 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh!");
        }
    }

    void tampil() {
        for (Mahasiswa11 m : listMhs) {
            m.tampil();
            System.out.println("-----");
        }
    }
}
```

Menambahkan 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) {
                Mahasiswa11 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}
```

Membuat class Main dan memberikan objek

```
public class Main11 {
    Run | Debug
    public static void main(String[] args) {
        DaftarMahasiswaBerprestasi11 list = new DaftarMahasiswaBerprestasi11();
        Mahasiswa11 m1 = new Mahasiswa11(n: "Nusa", t: 2017, u: 25, i: 3);
        Mahasiswa11 m2 = new Mahasiswa11(n: "Rara", t: 2012, u: 19, i: 4);
        Mahasiswa11 m3 = new Mahasiswa11(n: "Dompur", t: 2018, u: 19, i: 3.5);
        Mahasiswa11 m4 = new Mahasiswa11(n: "Abdul", t: 2017, u: 23, i: 2);
        Mahasiswa11 m5 = new Mahasiswa11(n: "Ummi", t: 2019, u: 21, i: 3.75);
    }
}
```

Memanggil fungsi tampil() dan fungsi bubbleSort()

```
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();
```

Hasil

```
c36827991872beb7ab5f23b5928\redh
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 = 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
-----
```

```
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 = Dompur
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
-----
```

5.2.3 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<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. Pada 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) {  
                Mahasiswa1 tmp = listMhs[j];  
                listMhs[j] = listMhs[j-1];  
                listMhs[j-1] = tmp;  
            }  
        }  
    }  
}
```

2. untuk menukar posisi ipk dari yang terbesar ke yang terkecil
3. a. Perulangan i digunakan untuk menentukan elemen array yang akan dibandingkan, sedangkan perulangan j digunakan untuk membandingkan elemen array dengan elemen yang ada disampingnya.
b. Karena index perulangan dimulai dari 0
c. Untuk menghindari pengecualian array out of bounds.
d. Pada perulangan i akan dilakukan perulangan sebanyak 50 kali dan setiap perulangan i akan melakukan tahap bubbleSort sebanyak 49 kali

Percobaan 5.3

Menambahkan method selectionSort() pada class 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;  
            }  
        }  
  
        Mahasiswa11 tmp = listMhs[idxMin];  
        listMhs[idxMin] = listMhs[i];  
        listMhs[i] = tmp;  
    }  
}
```

Memanggil method selectionSort pada class Main

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

Hasil

```
c36827991872beb7ab5f23b5928\redha
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
-----
```

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. Proses tersebut digunakan untuk menemukan indeks elemen dengan nilai IPK terkecil. Pertama melakukan inisialisasi idxMin; kemudian di dalam perulangan melakukan pengecekan apakah ipk pada elemen j lebih kecil dari ipk elemen idxMin.

Percobaan 5.4

Menambahkan method insertionSort() pada class DaftarMahasiswaBerprestasi

```
void insertionSort() {  
    for (int i = 1; i < listMhs.length; i++) {  
        Mahasiswa11 temp = listMhs[i];  
        int j = i;  
        while (j > 0 && listMhs[j-1].ipk > temp.ipk) {  
            listMhs[j] = listMhs[j-1];  
            j--;  
        }  
        listMhs[j] = temp;  
    }  
}
```

Memanggil method insertionSort() pada class Main

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

Hasil

```
c36827991872beb7ab5f23b5928\redha  
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 = 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  
-----
```

```
Data mahasiswa setelah sorting asc berdasarkan ipk  
Nama = Abdul  
Tahun masuk = 2017  
Umur = 23  
IPK = 2.0  
-----  
Nama = Nusa  
Tahun masuk = 2012  
Umur = 19  
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  
-----
```

5.4.3 Pertanyaan

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

Jawaban:

1. Dengan cara mengubah kondisi while menjadi (<)

```
void insertionSort() {  
    for (int i = 1; i < listMhs.length; i++) {  
        Mahasiswa11 temp = listMhs[i];  
        int j = i;  
        while (j > 0 && listMhs[j-1].ipk < temp.ipk) {  
            listMhs[j] = listMhs[j-1];  
            j--;  
        }  
        listMhs[j] = temp;  
    }  
}
```