

Praktikum 1: Mengimplementasikan Sorting menggunakan object

a. Bubble Sort

1. Buat class dengan nama Sorting(noAbsen), lalu isi atributnya dengan object array

```
public class Sorting07 {  
  
    int data[];  
    int jumData;
```

2. Lalu buat konstruktor dengan parameter int Data[] dan int jmlDat

```
Sorting07(int Data[], int jmlDat) {  
    jumData = jmlDat;  
    data = new int[jmlDat];  
    for (int i = 0; i < jumData; i++) {  
        data[i] = Data[i];  
    }  
}
```

3. Buatlah method void bubbleSort yang berisi algoritma sorting bubble sort

```
void bubbleSort() {  
    int temp = 0;  
    for (int i = 0; i < jumData - 1; i++) {  
        for (int j = 1; j < jumData - i; j++) {  
            if (data[j - 1] > data[j]) {  
                temp = data[j];  
                data[j] = data[j - 1];  
                data[j - 1] = temp;  
            }  
        }  
    }  
}
```

4. Lalu buat method void tampil untuk menampilkan sebelum dan sesudah sorting

```
void tampil() {  
    for (int i = 0; i < jumData; i++) {  
        System.out.print(data[i] + " ");  
    }  
    System.out.println();  
}
```

5. Buat class baru Bernama SortingMain(noAbsen) kemudian deklarasi array dengan nama a[] sekaligus dengan membuat objek dengan nama dataUrut1

```
int a[] = {20, 10, 2, 7, 12};  
Sorting07 dataUrut1 = new Sorting07(a, a.length);
```

6. Lalu panggil method bubbleSort dan tampil

```
System.out.println(x:"\nData Awal 1");
dataUrut1.tampil();
dataUrut1.bubbleSort();
System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
dataUrut1.tampil();
```

7. Hasil run program bubble sorting

```
Data Awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
```

b. Selection Sort

1. Pada class Sorting07 buat method void SelectionSort yang berisi algoritma Selection Sort

```
36     void SelectionSort() {
37         for (int i = 0; i < jumData - 1; i++) {
38             int min = i;
39             for (int j = i + 1; j < jumData; j++) {
40                 if (data[j] < data[min]) {
41                     min = j;
42                 }
43             }
44             int temp = data[i];
45             data[i] = data[min];
46             data[min] = temp;
47         }
48     }
```

2. Pada SortingMain07 deklarasikan array dengan nama b[] sekaligus dengan membuat objek dengan nama dataurut2

```
int b[] = {30, 20, 2, 8, 14};
Sorting07 dataUrut2 = new Sorting07(b, b.length);
```

3. Lakukan pemanggilan method tampil dan SelectionSort

```
18     System.out.println(x:"\nData Awal 2");
19     dataUrut2.tampil();
20     dataUrut2.SelectionSort();
21     System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
22     dataUrut2.tampil();
```

4. Hasil dari pemanggilan method SelectionSort dan tampil

```
Data Awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
```

c. Insertion Sort

1. Pada class Sorting07 buat method void InsertionSort yang berisi algoritma dari insertion sorting

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

2. Lalu pada SortingMain07 deklarasikan array c[] sekalian dengan membuat objek bernama dataUrut3

```
int c[] = {40, 10, 4, 9, 3};  
Sorting07 dataUrut3 = new Sorting07(c, c.length);
```

3. Lakukan pemanggilan method tampil dan InsertionSort

```
System.out.println(x: "\nData Awal 3");  
dataUrut3.tampil();  
dataUrut3.insertionSort();  
System.out.println(x: "Data sudah diurutkan dengan INSERTION SORT (ASC)");  
dataUrut3.tampil();
```

4. Hasil pemanggilan method tampil dan InsertionSort

```
Data Awal 3  
40 10 4 9 3  
Data sudah diurutkan dengan INSERTION SORT (ASC)  
3 4 9 10 40
```

6.2.5 Pertanyaan!

1. Jelaskan fungsi kode program berikut

Tim Ajar Algoritma dan Struktur Data 2024-2025
Jurusan Teknologi Informasi-Politeknik Negeri Malang

4

Algoritma dan Struktur Data 2024-2025



```
if (data[j-1]>data[j]){  
    temp=data[j];  
    data[j]=data[j-1];  
    data[j-1]=temp;  
}
```

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!
3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan

```
while (j>=0 && data[j]>temp)
```
4. Pada Insertion sort, apakah tujuan dari perintah `data[j+1]= data[j];`

Jawaban

1. Kode program tersebut digunakan untuk mengecek apakah nilai pada array data indeks j-1 lebih besar dari nilai pada array data indeks j, jika hasilnya true maka variabel temp akan menyimpan nilai dari array data pada indeks j, lalu array data indeks j akan menyimpan nilai dari array data indeks j-1, lalu array data indeks j-1 akan menyimpan nilai dari variabel temp. Yang intinya melakukan bubble sorting

```

void SelectionSort() {
    for (int i = 0; i < jumData - 1; i++) {
        int min = i;
        for (int j = i + 1; j < jumData; j++) {
            if (data[j] < data[min]) {
                min = j;
            }
        }
        int temp = data[i];
        data[i] = data[min];
        data[min] = temp;
    }
}

```

- 2.
3. Pada pengecekan kondisi tersebut harus memenuhi 2 syarat yaitu jika j memiliki nilai lebih dari sama dengan 0 dan jika nilai pada array data indeks j lebih besar dari nilai pada variabel temp. jadi jika nilai dari j adalah kurang dari 0, kondisi tidak terpenuhi
4. Untuk menyimpan nilai dari array data indeks j ke array data indeks j+1

Praktikum 2: Sorting Menurut IPK

1. Buatlah class Mahasiswa(noAbsen) beserta kosntruktor default, kosntruktor berparameter, dan method tampil

```

2  public class Mahasiswa07 {
3
4      String nim, nama, kelas;
5      double ipk;
6
7      Mahasiswa07() {
8      }
9
10     public Mahasiswa07(String nm, String name, String kls, double ip) {
11         nim = nm;
12         nama = name;
13         kelas = kls;
14         ipk = ip;
15     }
16
17     void tampilInformasi() {
18         System.out.println("Nama: " + nama);
19         System.out.println("NIM: " + nim);
20         System.out.println("Kelas: " + kelas);
21         System.out.println("IPK: " + ipk);
22     }
23 }
24

```

2. Lalu buat class MahasiswaBerprestasi(noAbsen)

```
public class MahasiswaBerprestasi07 {  
  
    Mahasiswa07 listMhs[] = new Mahasiswa07[5];  
    int idx;
```

3. Lalu tambahkan method tambah dan method tampil

```
void tambah(Mahasiswa07 m) {  
    if (idx < listMhs.length) {  
        listMhs[idx] = m;  
        idx++;  
    } else {  
        System.out.println(x:"data sudah penuh");  
    }  
}  
  
void tampil() {  
    for (Mahasiswa07 m : listMhs) {  
        m.tampilInformasi();  
        System.out.println(x:"-----");  
    }  
}
```

4. Lalu tambahkan 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) {  
                Mahasiswa07 tmp = listMhs[j];  
                listMhs[j] = listMhs[j - 1];  
                listMhs[j - 1] = tmp;  
            }  
        }  
    }  
}
```

5. Buat class MahasiswaDemo(noAbsen)

```
1 public class MahasiswaDemo07 {
    Run main | Debug main | Run | Debug
2     public static void main(String[] args) {
3         MahasiswaBerprestasi07 list = new MahasiswaBerprestasi07();
4         Mahasiswa07 m1 = new Mahasiswa07(nm:"123", name:"Zidan", kls:"2A", ip:3.2);
5         Mahasiswa07 m2 = new Mahasiswa07(nm:"124", name:"Ayu", kls:"2A", ip:3.5);
6         Mahasiswa07 m3 = new Mahasiswa07(nm:"125", name:"Sofi", kls:"2A", ip:3.1);
7         Mahasiswa07 m4 = new Mahasiswa07(nm:"126", name:"Sita", kls:"2A", ip:3.9);
8         Mahasiswa07 m5 = new Mahasiswa07(nm:"127", name:"Miki", kls:"2A", ip:3.7);
9
10        list.tambah(m1);
11        list.tambah(m2);
12        list.tambah(m3);
13        list.tambah(m4);
14        list.tambah(m5);
15
16        System.out.println(x:"Data mahasiswa sebelum sorting: ");
17        list.tampil();
18
19        System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK (DESC) : ");
20        list.bubbleSort();
21        list.tampil();
22    }
23 }
24
```

```
NIM: 123
Kelas: 2A
IPK: 3.2
-----
Nama: Ayu
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Sofi
NIM: 125
Kelas: 2A
IPK: 3.1
-----
Nama: Sita
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Miki
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Data Mahasiswa setelah sorting berdasarkan IPK (DESC) :
Nama: Sita
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Miki
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Ayu
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Zidan
NIM: 123
Kelas: 2A
IPK: 3.2
-----
Nama: Sofi
NIM: 125
Kelas: 2A
IPK: 3.1
-----
```

6.

6.3.4 Pertanyaan

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:

```
for (int i=0; i<listMhs.length-1; i++){  
    for (int j=1; j<listMhs.length-i; j++){
```

- a. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?
 - b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - c. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

Jawaban

1. bubbleSort
 - a. karena pengecekan dilakukan 4 kali berasal dari panjang array adalah 5.
 - b. Karena yang dibandingkan akan semakin sedikit
 - c. $i = 49$, tahap = 49

```

public class MahasiswaDemo07 {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi07 list = new MahasiswaBerprestasi07();
        Mahasiswa07 Mhs[] = new Mahasiswa07[list.listMhs.length];
        for (int i = 0; i < Mhs.length; i++) {
            if (i >= Mhs.length) {
                System.out.println(x:"Data sudah penuh");
            }else{
                System.out.println(x:"-----");
                System.out.println("Mahasiswa ke-" + (i+1));
                Mhs[i] = new Mahasiswa07();
                System.out.print(s:"Masukkan NIM: ");
                Mhs[i].nim = sc.nextLine();
                System.out.print(s:"Nama Mahasiswa: ");
                Mhs[i].nama = sc.nextLine();
                System.out.print(s:"Masukkan kelas: ");
                Mhs[i].kelas = sc.nextLine();
                System.out.print(s:"Masukkan IPK: ");
                Mhs[i].ipk = sc.nextDouble();
                sc.nextLine();
                list.tambah(Mhs[i]);
            }
        }

        System.out.println(x:"Data mahasiswa sebelum sorting: ");
        list.tampil();

        System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK (DESC) : ");
        list.bubbleSort();
        list.tampil();
    }
}

```

2.

Mahasiswa ke-1
Masukkan NIM: 123
Nama Mahasiswa: Zidan
Masukkan kelas: 2A
Masukkan IPK: 3,2

Mahasiswa ke-2
Masukkan NIM: 124
Nama Mahasiswa: Ayu
Masukkan kelas: 2A
Masukkan IPK: 3,5

Mahasiswa ke-3
Masukkan NIM: 125
Nama Mahasiswa: Sofi
Masukkan kelas: 2A
Masukkan IPK: 3,1

Mahasiswa ke-4
Masukkan NIM: 126
Nama Mahasiswa: Sita
Masukkan kelas: 2A
Masukkan IPK: 3,9

Mahasiswa ke-5
Masukkan NIM: 127
Nama Mahasiswa: Miki
Masukkan kelas: 2A
Masukkan IPK: 3,7

Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

1. Pada class MahasiswaBerprestasi lalu tambahkan method SelectionSort di dalamnya

```
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;  
            }  
        }  
        Mahasiswa07 tmp = listMhs[idxMin];  
        listMhs[idxMin] = listMhs[i];  
        listMhs[i] = tmp;  
    }  
}
```

2. Lalu panggil method SelectionSort dan tampil pada class MahasiswaDemo

```
System.out.println(x:"Data Mahasiswa setelah SELECTION SORT (ASC) berdasarkan IPK (DESC) : ");  
list.SelectionSort();  
list.tampil();
```

```

Data mahasiswa sebelum sorting:
Nama: Ali
NIM: 123
Kelas: 2B
IPK: 3.9
-----
Nama: ila
NIM: 124
Kelas: 2B
IPK: 3.1
-----
Nama: agus
NIM: 125
Kelas: 2B
IPK: 3.6
-----
Nama: tika
NIM: 126
Kelas: 2B
IPK: 3.3
-----
Nama: udin
NIM: 127
Kelas: 2B
IPK: 3.2
-----
Data Mahasiswa setelah SELECTION SORT (ASC) berdasarkan IPK (DESC) :
Nama: ila
NIM: 124
Kelas: 2B
IPK: 3.1
-----
Nama: udin
NIM: 127
Kelas: 2B
IPK: 3.2
-----
Nama: tika
NIM: 126
Kelas: 2B
IPK: 3.3
-----
Nama: agus
NIM: 125
Kelas: 2B
IPK: 3.6
-----
Nama: Ali
NIM: 123
Kelas: 2B
IPK: 3.9
-----
PS D:\Kuliah\Semester2\PRAKTIKUM-ASD\Jobsheet6>

```

3.

6.4.3 Pertanyaan

Di dalam method selection sort, terdapat baris program seperti di bawah ini:

```
int idxMin=i;
for (int j=i+1; j<listMhs.length; j++){
    if (listMhs[j].ipk<listMhs[idxMin].ipk){
        idxMin=j;
    }
}
```

Untuk apakah proses tersebut, jelaskan!

Jawaban

1. Untuk mencari nilai paling kecil diantara nilai yang ada dalam data array

Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

1. Pada class MahasiswaBerprestasi tambahkan method InsertionSort

```
void InsertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa07 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

2. Lalu pada class MahasiswaDemo panggil method InsertionSort dan tampil

```
System.out.println(x:"Data Mahasiswa setelah INSERTION SORT (ASC) berdasarkan IPK (DESC) : ");
list.InsertionSort();
list.tampil();
```

```

Data mahasiswa sebelum sorting:
Nama: ayu
NIM: 111
Kelas: 2c
IPK: 3.7
-----
Nama: dika
NIM: 222
Kelas: 2c
IPK: 3.0
-----
Nama: ila
NIM: 333
Kelas: 2c
IPK: 3.8
-----
Nama: susi
NIM: 444
Kelas: 2c
IPK: 3.1
-----
Nama: yayuk
NIM: 555
Kelas: 2c
IPK: 3.4
-----
Data Mahasiswa setelah INSERTION SORT (ASC) berdasarkan IPK (DESC) :
Nama: dika
NIM: 222
Kelas: 2c
IPK: 3.0
-----
Nama: susi
NIM: 444
Kelas: 2c
IPK: 3.1
-----
Nama: yayuk
NIM: 555
Kelas: 2c
IPK: 3.4
-----
Nama: ayu
NIM: 111
Kelas: 2c
IPK: 3.7
-----
Nama: ila
NIM: 333
Kelas: 2c
IPK: 3.8
-----

```

3.

6.5.3 Pertanyaan

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

Jawaban

Saya hanya mengubah kondisi pada while

```
void InsertionSort() {  
    for (int i = 1; i < listMhs.length; i++) {  
        Mahasiswa07 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 sebelum sorting:
Nama: w
NIM: 1
Kelas: 3,9
IPK: 3.9
-----
Nama: q
NIM: 2
Kelas: e
IPK: 3.0
-----
Nama: r
NIM: 4
Kelas: r
IPK: 3.4
-----
Nama: t
NIM: 3
Kelas: y
IPK: 3.1
-----
Nama: x
NIM: 6
Kelas: z
IPK: 3.3
-----
Data Mahasiswa setelah INSERTION SORT (ASC) berdasarkan IPK :
Nama: w
NIM: 1
Kelas: 3,9
IPK: 3.9
-----
Nama: r
NIM: 4
Kelas: r
IPK: 3.4
-----
Nama: x
NIM: 6
Kelas: z
IPK: 3.3
-----
Nama: t
NIM: 3
Kelas: y
IPK: 3.1
-----
Nama: q
NIM: 2
Kelas: e
IPK: 3.0
-----
```

Latihan Praktikum

```
1
2 public class Dosen {
3
4     String kode, nama;
5     boolean jenisKelamin; // true = laki, false perempuan
6     int usia;
7
8     Dosen(String kd, String name, boolean jk, int age) {
9         kode = kd;
10        nama = name;
11        jenisKelamin = jk;
12        usia = age;
13    }
14    Dosen() {}
15
16
17
18    void tampilInfoDosen() {
19        System.out.println("Kode: " + kode);
20        System.out.println("Nama: " + nama);
21        if (jenisKelamin) {
22            System.out.println(x: "Jenis Kelamin: Pria");
23        } else {
24            System.out.println(x: "Jenis Kelamin: Wanita");
25        }
26        System.out.println("Usia: " + usia);
27    }
28 }
29
```

```

1
2 public class DataDosen {
3
4     Dosen dataDosen[] = new Dosen[10];
5     int idx;
6
7     void tambahDosen(Dosen dsn) {
8         if (idx < dataDosen.length) {
9             dataDosen[idx] = dsn;
10            idx++;
11        } else {
12            System.out.println(x:"data sudah penuh");
13        }
14    }
15
16    void tampil() {
17        for (Dosen dsn : dataDosen) {
18            dsn.tampilInfoDosen();
19            System.out.println(x:"-----");
20        }
21    }
22
23    void SortingASC() {
24        for (int i = 0; i < dataDosen.length; i++) {
25            for (int j = 1; j < dataDosen.length - i; j++) {
26                if (dataDosen[j].usia > dataDosen[j - 1].usia) {
27                    Dosen temp = dataDosen[j];
28                    dataDosen[j] = dataDosen[j - 1];
29                    dataDosen[j - 1] = temp;
30                }
31            }
32        }
33    }
34
35    void SortingDSC() {
36        for (int i = 0; i < dataDosen.length - 1; i++) {
37            int idxMax = i;
38            for (int j = i + 1; j < dataDosen.length; j++) {
39                if (dataDosen[j].usia > dataDosen[idxMax].usia) {
40                    idxMax = j;
41                }
42            }
43            Dosen tmp = dataDosen[idxMax];
44            dataDosen[idxMax] = dataDosen[i];
45            dataDosen[i] = tmp;
46        }
47    }
48
49    void InsertionSort(){
50        for (int i = 1; i < dataDosen.length; i++) {
51            Dosen temp = dataDosen[i];
52            int j = i;
53            while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
54                dataDosen[j] = dataDosen[j - 1];
55                j--;
56            }
57            dataDosen[j] = temp;
58        }
59    }
60 }
61

```

```

1  import java.util.Scanner;
2  public class DosenMain {
3      Run main | Debug main | Run | Debug
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          DataDosen list = new DataDosen();
7          Dosen dsn[] = new Dosen[list.dataDosen.length];
8          boolean pilihan = true;
9          int choose;
10         String gender;
11
12         while (pilihan) {
13             System.out.println(x:"MENU DOSEN");
14             System.out.println(x:"1. Tambah Data Dosen");
15             System.out.println(x:"2. Tampil Informasi Seluruh Dosen");
16             System.out.println(x:"3. Sorting Ascending");
17             System.out.println(x:"4. Sorting Descending");
18             System.out.println(x:"0. Keluar");
19             System.out.print(s:"Pilihan: ");
20             choose = sc.nextInt();
21             sc.nextLine();
22             switch (choose) {
23                 case 0:
24                     pilihan = false;
25                     break;
26                 case 1:
27                     for (int i = 0; i < dsn.length; i++) {
28                         if (i >= dsn.length) {
29                             System.out.println(x:"Data sudah penuh");
30                         } else {
31                             System.out.println(x:"-----");
32                             System.out.println("Dosen ke-" + (i + 1));
33                             dsn[i] = new Dosen();
34                             System.out.print(s:"Masukkan Kode Dosen: ");
35                             dsn[i].kode = sc.nextLine();
36                             System.out.print(s:"Nama Dosen: ");
37                             dsn[i].nama = sc.nextLine();
38                             System.out.print(s:"Masukkan Jenis Kelamin(Pria/Wanita): ");
39                             gender = sc.nextLine();
40                             if (gender.equalsIgnoreCase(anotherString:"Pria")) {
41                                 dsn[i].jenisKelamin = true;
42                             } else if (gender.equalsIgnoreCase(anotherString:"Wanita")) {
43                                 dsn[i].jenisKelamin = false;
44                             } else {
45                                 System.out.println(x:"Gender salah, harp mengulang");
46                                 continue;
47                             }
48                             System.out.print(s:"Masukkan Usia: ");
49                             dsn[i].usia = sc.nextInt();
50                             sc.nextLine();
51                             list.tambahDosen(dsn[i]);
52                         }
53                     }
54                     break;
55                 case 2:
56                     list.tampil();
57                     break;
58                 case 3:
59                     list.SortingASC();
60                     list.tampil();
61                     break;
62                 case 4:
63                     list.SortingDSC();
64                     list.tampil();
65                     break;
66                 default:
67                     break;

```