

LAPORAN PRAKTIKUM

Oleh :

MUHAMMAD PEARL OCSHADA

NIM : 244107020064



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

1. Praktikum 1 - Mengimplementasikan Sorting menggunakan object

➤ Source Code

```
package Praktikum_ASD.ALSA_Jobsheet6;

public class sorting20 {
    int [] data;
    int jmlData;

    public sorting20(int Data[], int jmlData) {
        jmlData = jmlData;
        data = new int[jmlData];
        for (int i = 0; i < jmlData; i++) {
            data[i] = Data[i];
        }
    }

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

```

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

void insertionSort(){
    for (int i = 0; i < data.length; 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;
    }
}

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

```

```
package Praktikum_ASD.ALSD_Jobsheet6;

public class sortingMain20 {
    public static void main(String[] args) {
        int a[] = {20,10,2,7,12};
        sorting20 dataurut1 = new sorting20(a, a.length);

        System.out.println("Data Awal 1");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT");
        dataurut1.tampil();

        int b[] = {30,20,2,8,14};
        sorting20 dataurut2 = new sorting20(b,b.length);

        System.out.println("Data Awal 2");
        dataurut2.tampil();
        dataurut2.bubbleSort();
        System.out.println("Data sudah diurutkan dengan SELECTION SORT");
        dataurut2.tampil();

        int c[] = {40,10,4,9,3};
        sorting20 dataurut3 = new sorting20(c,c.length);

        System.out.println("Data Awal 3");
        dataurut3.tampil();
        dataurut3.bubbleSort();
        System.out.println("Data sudah diurutkan dengan INSERTION SORT");
        dataurut3.tampil();

    }
}
```

➤ Compile And Run

```
Data Awal 1
20 10 2 7 12
Data Sudah Diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20

Data Awal 2
30 20 2 8 14
Data Sudah Diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30

Data Awal 3
40 10 4 9 3
Data Sudah Diurutkan dengan INSERTION SORT (ASC)
3 4 9 10 40
PS D:\KuisDua07\Mata Kuliah\dewi\PrakDasPro_1G_07_De
```

➤ Pertanyaan :

1. Kode tersebut berfungsi untuk menukar dua elemen dalam array jika elemen sebelumnya lebih besar dari elemen saat ini dalam void Bubble Sort(), yang memastikan bahwa elemen yang lebih kecil dipindahkan ke posisi yang lebih tepat dalam array.
2. algoritma pencarian nilai minimum pada selection sort pada bagian :

```
int min = i;

    for (int j = i+1; j < jumData; j++) {

        if (data[j]<data[min]) {

            min = j;

        }

    }
```

- int min = i; (perumpamaan elemen di indeks i adalah yang terkecil dalam sisa array)
- for (int j = i + 1; j < jumData; j++) (Melakukan pencarian nilai terkecil dalam array mulai dari indeks i + 1 hingga jumData - 1.)
- if (data[j] < data[min]) (Membandingkan setiap elemen data[j] dengan data[min]. Jika ditemukan nilai lebih kecil, maka indeks min diperbarui ke j).

3. Kode ini digunakan digunakan untuk menggeser elemen ke kanan selama elemen sebelumnya lebih besar dari elemen yang ingin disisipkan (temp).
 - $j \geq 0$: Memastikan bahwa indeks j tidak menjadi negatif.
 - $data[j] > temp$: Mengecek apakah elemen $data[j]$ lebih besar dari elemen yang akan disisipkan (temp).Jika $data[j] > temp$, maka $data[j]$ digeser ke kanan ($data[j+1] = data[j]$).
Proses ini terus berulang sampai ditemukan posisi yang tepat untuk temp.
Setelah loop selesai, temp akan dimasukkan ke posisi yang benar ($data[j+1] = temp$).
4. Bertujuan untuk menggeser elemen yang lebih besar ke kanan agar elemen baru (temp) bisa dimasukkan ke posisi yang tepat.
 - Ketika elemen baru (temp) lebih kecil dari elemen sebelumnya ($data[j]$), maka $data[j]$ digeser ke $data[j+1]$.Ini dilakukan agar elemen temp bisa ditempatkan dalam posisi yang benar setelah elemen-elemen yang lebih besar digeser.

2. Praktikum 2- Sorting Menggunakan Array of Object

➤ Source Code

```
package Praktikum05;

public class mahasiswa20 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    public mahasiswa20() {

    }

    public mahasiswa20(String nm, String name, String kls, double ip) {

        nim = nm;

        nama = name;

        ipk = ip;

        kelas = kls;

    }

    void tampilInformasi(){

        System.out.println("Nama : " + nama);

        System.out.println("NIM : " + nim);

        System.out.println("IPK : " + ipk);

        System.out.println("Kelas : " + kelas);

    }

}
```

```

package Praktikum05;

public class mahasiswaBerprestasi {

    mahasiswa20 [] listMhs = new Mahasiswa07[5];

    int idx;

    void tambah (mahasiswa20 m){

        if (idx<listMhs.length) {

            listMhs[idx]=m;

            idx++;

        }else {

            System.out.println("data sudah penuh");

        }

    }

    void tampil(){
        for(mahasiswa20 m :listMhs){
            if (m != null) {
                m.tampilInformasi();
            }
        }
        -----

    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;

                }

            }

        }

    }

}

```



```
package Praktikum05;

public class mahasiswaDemo20 {

    public static void main(String[] args) {

        mahasiswaBerprestasi07 list = new mahasiswaBerprestasi();

        mahasiswa20 m1 = new mahasiswa07("123", "Zidan", "2A", 3.2);
        mahasiswa20 m2 = new mahasiswa07("124", "Ayu", "2A", 3.5);
        mahasiswa20 m3 = new mahasiswa07("125", "Sofi", "2A", 3.1);
        mahasiswa20 m4 = new mahasiswa07("126", "Sita", "2A", 3.9);
        mahasiswa20 m5 = new mahasiswa07("127", "Miki", "2A", 3.7);

        list.tambah(m1);
        list.tambah(m2);
        list.tambah(m3);
        list.tambah(m4);
        list.tambah(m5);

        System.out.println("Data Mahasiswa Sebelum Sorting : ");
        list.tampil();

        System.out.println("Data Mahasiswa Sesudah Sorting Berdasarkan IPK (DESC) : ");
        list.bubbleSort();
        list.tampil();

    }

}
```

➤ Compile And Run

```
Data Mahasiswa Sebelum Sorting :
Nama : Zidan
NIM : 123
IPK : 3.2
Kelas : 2A
-----
Nama : Ayu
NIM : 124
IPK : 3.5
Kelas : 2A
-----
Nama : Sofi
NIM : 125
IPK : 3.1
Kelas : 2A
-----
Nama : Sita
NIM : 126
IPK : 3.9
Kelas : 2A
-----
Nama : Miki
NIM : 127
IPK : 3.7
Kelas : 2A
-----
```

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

➤ Pertanyaan :

1.

a) .

Karena :

- i merupakan jumlah iterasi utama dalam Bubble Sort.
- Setiap iterasi, elemen terbesar dari sisa array akan ditempatkan di posisi yang benar.
- kita hanya perlu melakukan N-1 iterasi untuk memastikan semua elemen sudah terurut, maka~ batasnya adalah listMhs.length - 1.
- Jika i mencapai listMhs.length - 1, maka array sudah terurut sepenuhnya.

b) .

karena :

- j digunakan untuk membandingkan dan menukar elemen bersebelahan dalam array.
- Setiap iterasi i, satu elemen terbesar sudah berada di posisi yang benar.
- Maka, bagian array yang perlu di-sorting berkurang sebanyak i, sehingga batas perulangan dalam j adalah listMhs.length - i.

c) .

Jika banyak data dalam listMhs adalah 50, maka:

- i akan berjalan dari 0 hingga 49 (total 49 kali perulangan).
- Bubble Sort selalu membutuhkan N-1 tahap, sehingga jumlah tahap = 49 tahap.

2. Modifikasi Program agar Data Mahasiswa Diinput Secara Dinamis

```
package Praktikum05;  
  
import java.util.Scanner;  
  
public class mahasiswaDemo20 {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        mahasiswaBerprestasi list = new mahasiswaBerprestasi();  
  
        System.out.println("Data Mahasiswa Sebelum Sorting : ");  
        list.tampil();  
  
        for (int i = 0; i < 5; i++) {  
  
            System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));  
  
            System.out.print("NIM      : ");  
  
            String nim = sc.nextLine();  
  
            System.out.print("Nama    : ");  
  
            String nama = sc.nextLine();  
  
            System.out.print("Kelas  : ");  
  
            String kelas = sc.nextLine();  
  
            System.out.print("IPK     : ");  
  
            String ipkStr = sc.nextLine().replace(",", ".");  
  
            double ipk = Double.parseDouble(ipkStr);  
  
            Mahasiswa20 m = new mahasiswa20(nim, nama, kelas, ipk);  
  
            list.tambah(m);  
  
            System.out.println("-----");  
        }  
    }  
}
```

```

        }

        System.out.println();

        System.out.println("Data Mahasiswa Sesudah Sorting Berdasarkan IPK
(DESC) : ");

        list.bubbleSort();

        list.tampil();

    }
}

```

```

package Praktikum05;

public class MahasiswaBerprestasi07 {

    mahasiswa07 [] listMhs = new mahasiswa20[5];

    int idx;

    void tambah (mahasiswa20 m){
        if (idx<listMhs.length) {
            listMhs[idx]=m;
            idx++;
        }else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil (){
        for (mahasiswa20 m:listMhs) {
            if (m != null) {
                m.tampilInformasi();
            }
        }
    }
}

```

```

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) {
                mahasiswa20 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}

```

Data Mahasiswa Sebelum Sorting :

Masukkan Data Mahasiswa ke-1

NIM : 123

Nama : Ali

Kelas : 2B

IPK : 3.9

Masukkan Data Mahasiswa ke-2

NIM : 124

Nama : ila

Kelas : 2B

IPK : 3.1

Masukkan Data Mahasiswa ke-3

NIM : 125

Nama : agus

Kelas : 3.6

IPK : 3.6

Masukkan Data Mahasiswa ke-4

NIM : 126

Nama : Tika

Kelas : 2B

IPK : 3.3

Masukkan Data Mahasiswa ke-5

NIM : 127

Nama : udin

Kelas : 2B

IPK : 3.2

Data Mahasiswa Sesudah Sorting Berdasarkan IPK (DESC)

NIM : 123

Nama : Ali

Kelas : 2B

IPK : 3.9

NIM : 125

Nama : agus

Kelas : 3.6

IPK : 3.6

NIM : 126

Nama : Tika

Kelas : 2B

IPK : 3.3

NIM : 127

Nama : udin

Kelas : 2B

IPK : 3.2

NIM : 124

Nama : ila

Kelas : 2B

IPK : 3.1

PS D:\KuisDua07\Mata Kuliah\dewi\PrakDasPro_16_07_Dewi

3. Praktikum 3- Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

➤ Source Code

```
package Praktikum05;  
  
public class mahasiswaBerprestasi {  
    mahasiswa20 [] listMhs = new mahasiswa20[5];  
  
    int idx;  
  
    void tambah (mahasiswa20 m){  
        if (idx<listMhs.length) {  
            listMhs[idx]=m;  
            idx++;  
        }else {  
            System.out.println("data sudah penuh");  
        }  
    }  
  
    void tampil (){  
        for (mahasiswa20 m:listMhs) {  
            if (m != null) {  
                m.tampilInformasi();  
            }  
        }  
    }  
  
    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) {  
                    mahasiswa20 tmp = listMhs[j];  
                    listMhs[j] = listMhs[j-1];  
                    listMhs[j-1] = tmp;  
                }  
            }  
        }  
    }  
}
```

```

    }

}

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) {
                mahasiswa20 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;
            }
        }
        mahasiswa07 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
}

```

```

package Praktikum05;

import java.util.Scanner;

public class mahasiswaDemo20 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        mahasiswaBerprestasi list = new mahasiswaBerprestasi();

        System.out.println("Data Mahasiswa Sebelum Sorting : ");
        list.tampil();

        for (int i = 0; i < 5; i++) {

            System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));

            System.out.print("NIM      : ");
            String nim = sc.nextLine();

            System.out.print("Nama    : ");
            String nama = sc.nextLine();

            System.out.print("Kelas  : ");
            String kelas = sc.nextLine();

            System.out.print("IPK     : ");
            String ipkStr = sc.nextLine().replace(",", ".");

            double ipk = Double.parseDouble(ipkStr);

            mahasiswa20 m = new mahasiswa20(nim, nama, kelas, ipk);
            list.tambah(m);

            System.out.println("-----");
        }

        System.out.println();

        System.out.println("Data Mahasiswa Sesudah Sorting Berdasarkan IPK (DESC) : ");
        list.bubbleSort();
        list.tampil();

        System.out.println();

        System.out.println("Data Yang Sudah Terurut Menggunakan SELECTION SORT (ASC)");
        list.selectionSort();
        list.tampil();

    }
}

```


➤ Compile And Run

```
Data Mahasiswa Sebelum Sorting :
Masukkan Data Mahasiswa ke-1
NIM : 123
Nama : Ali
Kelas : 2B
IPK : 3.9
-----
Masukkan Data Mahasiswa ke-2
NIM : 124
Nama : ila
Kelas : 2B
IPK : 3.1
-----
Masukkan Data Mahasiswa ke-3
NIM : 125
Nama : agus
Kelas : 2B
IPK : 3.6
-----
Masukkan Data Mahasiswa ke-4
NIM : 126
Nama : tika
Kelas : 2B
IPK : 3.3
-----
Masukkan Data Mahasiswa ke-5
NIM : 127
Nama : udin
Kelas : 2B
IPK : 3.2
-----

Data Yang Sudah Terurut Menggunakan SELECTION SORT (ASC)
NIM : 124
Nama : ila
Kelas : 2B
IPK : 3.1
-----
NIM : 127
Nama : udin
Kelas : 2B
IPK : 3.2
-----
NIM : 126
Nama : tika
Kelas : 2B
IPK : 3.3
-----
NIM : 125
Nama : agus
Kelas : 2B
IPK : 3.6
-----
NIM : 123
Nama : Ali
Kelas : 2B
IPK : 3.9
-----
```

➤ Pertanyaan:

1. Bagian proses tersebut digunakan untuk menentukan indeks elemen dengan nilai IPK terkecil dalam sisa array, untuk mengurutkan data berdasarkan nilai IPK mahasiswa secara ascending (dari kecil ke besar).

Inisialisasi indeks minimum (idxMin)

- idxMin = i;

- Mengasumsikan bahwa elemen pada indeks i adalah yang terkecil.

Mencari elemen terkecil dalam sisa array

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

- Loop ini memeriksa elemen mulai dari i+1 hingga akhir array.

Memperbarui indeks jika ditemukan elemen lebih kecil

- if (listMhs[j].ipk < listMhs[idxMin].ipk) { idxMin = j; }

- Jika ditemukan IPK yang lebih kecil dari idxMin, maka indeks idxMin diperbarui ke j.

4. Praktikum 4- Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

➤ Source Code

```
package Praktikum05;

public class mahasiswaBerprestasi {
    mahasiswa20 [] listMhs = new mahasiswa20[5];
    int idx;

    void tambah (mahasiswa20 m){
        if (idx<listMhs.length) {
            listMhs[idx]=m;
            idx++;
        }else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil (){
        for (mahasiswa20 m:listMhs) {
            if (m != null) {
                m.tampilInformasi();
            }
        }
    }

    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) {
                    mahasiswa20 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;
            }
        }
        Mahasiswa20 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}

void insertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        mahasiswa20 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
}

```

```
package Praktikum05;  
  
import java.util.Scanner;  
  
public class mahasiswaDemo20 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        mahasiswaBerprestasi list = new mahasiswaBerprestasi();  
  
        System.out.println("Data Mahasiswa Sebelum Sorting : ");  
        list.tampil();  
  
        for (int i = 0; i < 5; i++) {  
            System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));  
            System.out.print("NIM      : ");  
            String nim = sc.nextLine();  
            System.out.print("Nama    : ");  
            String nama = sc.nextLine();  
            System.out.print("Kelas : ");  
            String kelas = sc.nextLine();  
            System.out.print("IPK      : ");  
            String ipkStr = sc.nextLine().replace(",", ".");  
            double ipk = Double.parseDouble(ipkStr);  
  
            mahasiswa20 m = new mahasiswa20(nim, nama, kelas, ipk);  
            list.tambah(m);  
            System.out.println("-----");  
        }  
  
        System.out.println();  
        System.out.println("Data Mahasiswa Sesudah Sorting Berdasarkan IPK (DESC) : ");  
        list.bubbleSort();  
        list.tampil();  
    }  
}
```

```

        System.out.println();

        System.out.println("Data Yang Sudah Terurut Menggunakan SELECTION SORT (ASC)");
        list.selectionSort();

        list.tampil();

        System.out.println();

        System.out.println("Data Yang Sudah Terurut Menggunakan INSERTION SORT (ASC)");
        list.insertionSort();

        list.tampil();
    }
}

```

➤ Compile And Run

```

Data Mahasiswa Sebelum Sorting :
Masukkan Data Mahasiswa ke-1
NIM : 111
Nama : ayu
Kelas : 2c
IPK : 3.7
-----
Masukkan Data Mahasiswa ke-2
NIM : 222
Nama : dika
Kelas : 2c
IPK : 3.0
-----
Masukkan Data Mahasiswa ke-3
NIM : 333
Nama : ila
Kelas : 2c
IPK : 3.8
-----
Masukkan Data Mahasiswa ke-4
NIM : 444
Nama : susi
Kelas : 2c
IPK : 3.1
-----
Masukkan Data Mahasiswa ke-5
NIM : 555
Nama : yayuk
Kelas : 2c
IPK : 3.4
-----

```

```

Data Yang Sudah Terurut Menggunakan INSERTION SORT (ASC)
NIM : 222
Nama : dika
Kelas : 2c
IPK : 3.0
-----
NIM : 444
Nama : susi
Kelas : 2c
IPK : 3.1
-----
NIM : 555
Nama : yayuk
Kelas : 2c
IPK : 3.4
-----
NIM : 111
Nama : ayu
Kelas : 2c
IPK : 3.7
-----
NIM : 333
Nama : ila
Kelas : 2c
IPK : 3.8
-----

```

➤ **Pertanyaan :**

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

Solusi:

Perubahan yang dilakukan adalah pada fungsi insertionSort() pada bagian while Tanda perbandingan diubah → dari > menjadi <

```
void insertionSort(){
    for (int i = 1; i <
listMhs.length; i++) {
        mahasiswa20 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 :
Masukkan Data Mahasiswa ke-1
NIM : 111
Nama : ayu
Kelas : 2c
IPK : 3.7
-----
Masukkan Data Mahasiswa ke-2
NIM : 222
Nama : dika
Kelas : 2c
IPK : 3.0
-----
Masukkan Data Mahasiswa ke-3
NIM : 333
Nama : ila
Kelas : 2c
IPK : 3.8
-----
Masukkan Data Mahasiswa ke-4
NIM : 444
Nama : yayuk
Kelas : 2c
IPK : 3.1
-----
Masukkan Data Mahasiswa ke-5
NIM : 555
Nama : susi
Kelas : 2c
IPK : 3.4
-----
```

```
Data Yang Sudah Terurut Menggunakan INSERTION SORT (DESC)
NIM : 333
Nama : ila
Kelas : 2c
IPK : 3.8
-----
NIM : 111
Nama : ayu
Kelas : 2c
IPK : 3.7
-----
NIM : 555
Nama : susi
Kelas : 2c
IPK : 3.4
-----
NIM : 444
Nama : yayuk
Kelas : 2c
IPK : 3.1
-----
NIM : 222
Nama : dika
Kelas : 2c
IPK : 3.0
-----
```

❖ Latihan Praktikum

➤ Source Code

```
package Praktikum_ASD.ALSA_JobSheet6;

public class dosen {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;
    dosen(String kd, String name, boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

    public void tampil() {
        System.out.println();
        System.out.println("Kode          : " + kode);
        System.out.println("Nama          : " + nama);
        System.out.println("Jenis Kelamin : " + (jenisKelamin ? "Laki-laki" : "Perempuan"));
        System.out.println("Usia          : " + usia);
        System.out.println("-----");
    }
}
```

```
package Praktikum_ASD.ALSD_Jobsheet6;

public class dataDosen {
    dosen[] dataDosen = new dosen[10];
    int idx = 0;
    void tambah(dosen dsn){
        if (idx < dataDosen.length) {
            dataDosen[idx]=dsn;
            idx++;
        }else {
            System.out.println("data sudah penuh");
        }
    }
    void tampil(){
        if (idx == 0) {
            System.out.println("Belum ada data");
        } else {
            for (int i = 0; i < idx; i++) {
                dataDosen[i].tampil();
            }
        }
    }
    void sortingASC(){
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - i - 1; j++) {
                if (dataDosen[j].usia > dataDosen[j + 1].usia) {
                    dosen temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
    }
}
```



```

void sortingDSC(){
    for (int i = 0; i < idx - 1; i++) {
        int idxMax = i;
        for (int j = i + 1; j < idx; j++) {
            if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                idxMax = j;
            }
        }
        dosen tmp = dataDosen[idxMax];
        dataDosen[idxMax] = dataDosen[i];
        dataDosen[i] = tmp;
    }
}

void insertionSort(){
    for (int i = 1; i < idx; i++) {
        dosen temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
}
}

```

```
package Praktikum05;

import java.util.Scanner;

public class DosenMain07 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        DataDosen07 listDosen = new DataDosen07();

        while (true) {

            System.out.println("===== Menu =====");

            System.out.println("1. Tambahkan Data Dosen");

            System.out.println("2. Tampilkan Data Dosen");

            System.out.println("3. Urutkan Data Bubble Sort (ASC)");

            System.out.println("4. Urutkan Data Selection Sort (DSC)");

            System.out.println("5. Urutkan Data Insertion Sort (DSC)");

            System.out.println("0. Keluar");

            System.out.print("Pilih menu: ");

            int pilihan = sc.nextInt();

            sc.nextLine();

            switch (pilihan) {

                case 1:

                    System.out.println();

                    for (int i = 0; i < 10; i++) {

                        System.out.print("Kode : ");

                        String kode = sc.nextLine();

                        System.out.print("Nama : ");

                        String nama = sc.nextLine();

                        System.out.print("Jenis Kelamin (true = Laki-laki, false = Perempuan) : ");

                        boolean jk = sc.nextBoolean();

                        System.out.print("Usia : ");

                        int usia = sc.nextInt();

                        sc.nextLine();

                        System.out.println();

                        Dosen07 dsn = new Dosen07(kode, nama, jk, usia);

                        listDosen.tambah(dsn);
```

```

        }

        break;

    case 2:

        System.out.println();

        System.out.println("=== Data Keseluruhan ===");

        listDosen.tampil();

        break;

    case 3:

        System.out.println();

        System.out.println("Data telah diurutkan (ASC - Bubble Sort).");

        listDosen.sortingASC();

        listDosen.tampil();

        break;

    case 4:

        System.out.println();

        System.out.println("Data telah diurutkan (DSC - Selection Sort).");

        listDosen.sortingDSC();

        listDosen.tampil();

        break;

    case 5:

        System.out.println();

        System.out.println("Data telah diurutkan (DSC - Insertion Sort).");

        listDosen.insertionSort();

        listDosen.tampil();

        break;

    case 0:

        System.out.println("Keluar dari program.");

        return;

    default:

        System.out.println("Pilihan tidak valid.");

}

}

}

}

```

➤ Compile And Run

```
===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 1

Kode : 01
Nama : Dewi
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 20

Kode : 02
Nama : Ambar
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 21

Kode : 03
Nama : Anthea
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 22

Kode : 04
Nama : Ayun
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 23

Kode : 05
Nama : Salma
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 24
```

```
Kode : 06
Nama : Fira
Jenis Kelamin (true = Laki-laki, false = Perempuan) : false
Usia : 25

Kode : 07
Nama : Pearl
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 26

Kode : 08
Nama : Aldy
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 27

Kode : 09
Nama : Arya
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 28

Kode : 10
Nama : Bagus
Jenis Kelamin (true = Laki-laki, false = Perempuan) : true
Usia : 29
```

```
Pilih menu: 2

=== Data Keseluruhan ===

Kode      : 01
Nama      : Dewi
Jenis Kelamin : Perempuan
Usia      : 20
-----

Kode      : 02
Nama      : Ambar
Jenis Kelamin : Perempuan
Usia      : 21
-----

Kode      : 03
Nama      : Anthea
Jenis Kelamin : Perempuan
Usia      : 22
-----

Kode      : 04
Nama      : Ayun
Jenis Kelamin : Perempuan
Usia      : 23
-----

Kode      : 05
Nama      : Salma
Jenis Kelamin : Perempuan
Usia      : 24
-----
```

```
-----
Kode      : 06
Nama      : Fira
Jenis Kelamin : Perempuan
Usia      : 25
-----

Kode      : 07
Nama      : Pearl
Jenis Kelamin : Laki-laki
Usia      : 26
-----

Kode      : 08
Nama      : Aldy
Jenis Kelamin : Laki-laki
Usia      : 27
-----

Kode      : 09
Nama      : Arya
Jenis Kelamin : Laki-laki
Usia      : 28
-----

Kode      : 10
Nama      : Bagus
Jenis Kelamin : Laki-laki
Usia      : 29
-----
```

```
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 3

Data telah diurutkan (ASC - Bubble Sort).

Kode      : 01
Nama      : Dewi
Jenis Kelamin : Perempuan
Usia      : 20
-----

Kode      : 02
Nama      : Ambar
Jenis Kelamin : Perempuan
Usia      : 21
-----

Kode      : 03
Nama      : Anthea
Jenis Kelamin : Perempuan
Usia      : 22
-----

Kode      : 04
Nama      : Ayun
Jenis Kelamin : Perempuan
Usia      : 23
-----
```

```
Kode      : 05
Nama      : Salma
Jenis Kelamin : Perempuan
Usia      : 24
-----

Kode      : 06
Nama      : Fira
Jenis Kelamin : Perempuan
Usia      : 25
-----

Kode      : 07
Nama      : Pearl
Jenis Kelamin : Laki-laki
Usia      : 26
-----

Kode      : 08
Nama      : Aldy
Jenis Kelamin : Laki-laki
Usia      : 27
-----

Kode      : 09
Nama      : Arya
Jenis Kelamin : Laki-laki
Usia      : 28
-----

Kode      : 10
Nama      : Bagus
Jenis Kelamin : Laki-laki
```

```

Kode      : 08
Nama      : Aldy
Jenis Kelamin : Laki-laki
Usia      : 27
-----

Kode      : 07
Nama      : Pearl
Jenis Kelamin : Laki-laki
Usia      : 26
-----

Kode      : 06
Nama      : Fira
Jenis Kelamin : Perempuan
Usia      : 25
-----

Kode      : 05
Nama      : Salma
Jenis Kelamin : Perempuan
Usia      : 24
-----

Kode      : 04
Nama      : Ayun
Jenis Kelamin : Perempuan
Usia      : 23
-----

Kode      : 03
Nama      : Anthea

```

```

-----
Kode      : 03
Nama      : Anthea
Jenis Kelamin : Perempuan
Usia      : 22
-----

```

```

Kode      : 02
Nama      : Ambar
Jenis Kelamin : Perempuan
Usia      : 21
-----

```

```

Kode      : 01
Nama      : Dewi
Jenis Kelamin : Perempuan
Usia      : 20
-----

```

```

===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 0
Keluar dari program.
PS D:\KuisDua07\Mata Kuliah\dewi\PrakDa

```

```

Jenis Kelamin : Laki-laki
Usia      : 29
-----
===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 4

Data telah diurutkan (DSC - Selection Sort).

Kode      : 10
Nama      : Bagus
Jenis Kelamin : Laki-laki
Usia      : 29
-----

Kode      : 09
Nama      : Arya
Jenis Kelamin : Laki-laki
Usia      : 28
-----

Kode      : 08
Nama      : Aldy
Jenis Kelamin : Laki-laki
Usia      : 27
-----

Kode      : 07

```

```

Kode      : 07
Nama      : Pearl
Jenis Kelamin : Laki-laki
Usia      : 26
-----

Kode      : 06
Nama      : Fira
Jenis Kelamin : Perempuan
Usia      : 25
-----

Kode      : 05
Nama      : Salma
Jenis Kelamin : Perempuan
Usia      : 24
-----

Kode      : 04
Nama      : Ayun
Jenis Kelamin : Perempuan
Usia      : 23
-----

Kode      : 03
Nama      : Anthea
Jenis Kelamin : Perempuan
Usia      : 22
-----

Kode      : 02
Nama      : Ambar
Jenis Kelamin : Perempuan

```

```

Jenis Kelamin : Perempuan
Usia      : 21
-----

Kode      : 01
Nama      : Dewi
Jenis Kelamin : Perempuan
Usia      : 20
-----
===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 5

Data telah diurutkan (DSC - Insertion Sort).

Kode      : 10
Nama      : Bagus
Jenis Kelamin : Laki-laki
Usia      : 29
-----

Kode      : 09
Nama      : Arya
Jenis Kelamin : Laki-laki
Usia      : 28
-----

Kode      : 08

```

