

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DASAR
JOBSHEET 6



NAMA : DANDIKA MARTHA C.
NIM : 244107020092
KELAS : 1E

Program Studi Teknik Informatika
Jurusan Teknologi Informasi
Praktikum
2025

Percobaan 1

Sorting07.java

```
package jobsheet_6;

public class Sorting07 {

    int [] data;
    int jumData;

    Sorting07 (int Data[], int jmlDat){
        jumData=jmlDat;
        data=new int[jmlDat];
        for (int i=0; i<jumData; i++) {
            data[i]= Data[i];
        }
    }
    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;
                }
            }
        }
    }
    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;
        }
    }
    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;
        }
    }
    void tampil(){
        for (int i=0; i<jumData;i++){
            System.out.print(data[i]+" ");
        }
        System.out.println();
    }
}
```

SortingMain07.java

```
package jobsheet_6;

public class SortingMain07 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        Sorting07 dataurut1 = new Sorting07(a, a.length);
        System.out.println("Data awal 1");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataurut1.tampil();

        int b[] = {30, 20, 2, 8, 14};
        Sorting07 dataurut2 = new Sorting07(b, b.length);
        System.out.println("Data awal 2");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println("Data sudah diurutkan dengan SELECTION SORT (ASC)");
        dataurut2.tampil();

        int c[] = {40, 10, 4, 9, 3};
        Sorting07 dataurut3 = new Sorting07(c, c.length);
        System.out.println("Data awal 3");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println("Data sudah diurutkan dengan INSERTION SORT (ASC)");
        dataurut3.tampil();
    }
}
```

Hasil

- Bubble Sort

```
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>
```

- Selection Sort

```
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>
```

- Insertion Sort

```
Data awal 3
40 10 4 9 3
Data sudah diurutkan dengan INSERTION SORT (ASC)
4 9 10 40 3
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>
```

Pertanyaan

1. Jelaskan fungsi kode program berikut

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

Jawab : Kode tersebut untuk memindahkan nilai apabila, nilai sebelah kiri lebih besar dari sebelah kanan. Temp berisi nilai dari indeks data ke-j, dan untuk data indeks ke-j menyimpan data indeks ke-j-1. Terakhir data indeks ke-j-1 menyimpan data dari temp.

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!

Jawab : i = 0

```
int min=i;  
for (int j=i+1; j<jumData; j++) {  
    if(data[j]<data[min]) {  
        min=j;  
    }  
}
```

3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan

```
while (j>=0 && data[j]>temp)
```

Jawab : j>=0 untuk memastikan bahwa indeks yang di akses adalah positif dan untuk **data[j]>temp** memastikan bahwa kita hanya memindahkan elemen yang lebih besar dari temp ke kanan. Selama elemen yang ada di indeks j lebih besar dari temp, kita akan memindahkan elemen data[j] ke posisi data[j+1].

4. Pada Insertion sort, apakah tujuan dari perintah

```
data[j+1]=data[j];
```

Jawab : Perintah `data[j + 1] = data[j];` menggeser elemen-elemen yang lebih besar dari temp (nilai yang sedang disisipkan) ke kanan, sehingga memberi ruang untuk menempatkan temp di posisi yang tepat dalam array.

Percobaan 2

Mahasiswa07.java

```
package jobsheet_6;

public class Mahasiswa07 {
    String nama;
    String nim;
    String kelas;
    double ipk;
    public Mahasiswa07() {
    }
    public Mahasiswa07 (String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        ipk = ip;
        kelas = kls;
    }
    void tampilkanInformasi () {
        System.out.println("Nama      : " + nama);
        System.out.println("NIM      : " + nim);
        System.out.println("Kelas   : " + kelas);
        System.out.println("IPK      : " + ipk);
    }
}
```

MahasiswaBerprestasi07.java

```
package jobsheet_6;

public class MahasiswaBerprestasi07 {
    Mahasiswa07 [] listMhs = new Mahasiswa07[5];
    int idx;
    void tambah (Mahasiswa07 m) {
        if (idx<listMhs.length) {
            listMhs[idx]=m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }
    void tampil() {
        for (Mahasiswa07 m:listMhs){
            m.tampilkanInformasi();
            System.out.println("-----");
        }
    }
    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;
                }
            }
        }
    }
}
```

MahasiswaDemo07.java

```
package jobsheet_6;

public class MahasiswaDemo07 {
    public static void main(String[] args) {
        MahasiswaBerprestasi07 list = new MahasiswaBerprestasi07();
        Mahasiswa07 m1 = new Mahasiswa07("123", "Zidan", "2A", 3.2);
        Mahasiswa07 m2 = new Mahasiswa07("124", "Ayu", "2A", 3.5);
        Mahasiswa07 m3 = new Mahasiswa07("125", "Sofi", "2A", 3.1);
        Mahasiswa07 m4 = new Mahasiswa07("126", "Sita", "2A", 3.9);
        Mahasiswa07 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 setelah sorting berdasarkan
IPK (DESC) : ");
        list.bubbleSort();
        list.tampil();
    }
}
```

Hasil

```
Data mahasiswa sebelum sorting:
Nama   : Zidan
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
-----
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>
```

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 $i < \text{listMhs.length} - 1$?

Jawab : Karena pada iterasi pertama kita sudah mendapatkan nilai terbesar atau terkecil di posisi terakhir dan itu sudah pasti benar, sehingga kita tidak perlu melakukan iterasi sesuai dengan banyak elemen.

b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?

Jawab : Karena kode tersebut digunakan untuk mengurangi perbandingan antara nilai indeks. Karena pada iterasi pertama indeks terakhir pasti berisi nilai terbesar atau terkecil.

c. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Jawab : Perulangan i terjadi dari $i = 0$ sampai $i = 48$ atau sebanyak 49 kali. Tahap yang ditempuh bubble sort sebanyak 49 kali.

2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

```
package jobsheet_6;
import java.util.Scanner;
public class MahasiswaDemo07 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi07 list = new MahasiswaBerprestasi07();
        for(int i=0; i < 5; i++){
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));
            System.out.print("Masukkan NIM      : ");
            String nim = sc.nextLine();
            System.out.print("Masukkan Nama      : ");
            String nama = sc.nextLine();
            System.out.print("Masukkan Kelas      : ");
            String kelas = sc.nextLine();
            System.out.print("Masukkan IPK      : ");
            String dummy = sc.nextLine();
            double ipk = Double.parseDouble(dummy);
            System.out.println("-----");
            Mahasiswa07 mahasiswa = new Mahasiswa07(nim, nama, kelas,
            ipk);
            list.tambah(mahasiswa);
        }
        System.out.println("Data mahasiswa sebelum sorting: ");
        list.tampil();

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


Hasil

```
Masukkan data mahasiswa ke-1
Masukkan NIM      : 123
Masukkan Nama     : Zidan
Masukkan Kelas    : 2A
Masukkan IPK      : 3.2
```

```
-----
Masukkan data mahasiswa ke-2
Masukkan NIM      : 124
Masukkan Nama     : Ayu
Masukkan Kelas    : 2A
Masukkan IPK      : 3.5
```

```
-----
Masukkan data mahasiswa ke-3
Masukkan NIM      : 125
Masukkan Nama     : Sofi
Masukkan Kelas    : 2A
Masukkan IPK      : 3.1
```

```
-----
Masukkan data mahasiswa ke-4
Masukkan NIM      : 126
Masukkan Nama     : Sita
Masukkan Kelas    : 2A
Masukkan IPK      : 3.9
```

```
-----
Masukkan data mahasiswa ke-5
Masukkan NIM      : 127
Masukkan Nama     : Miki
Masukkan Kelas    : 2A
Masukkan IPK      : 3.7
```

Percobaan 3

- Tambahkan kode pada MahasiswaBerprestasi07.java

```
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;
    }
}
```

- Tambahkan kode pada MahasiswaDemo07.java

```
System.out.println("Data yang sudah terurut menggunakan SELECTION SORT (ASC)");
list.selectionSort();
list.tampil();
```

Hasil

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

```

-----
Data yang sudah terurut menggunakan SELECTION SORT (ASC)
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 G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>

```

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!

Jawab : **idxMin** digunakan untuk nilai awal yang akan di bandingkan dengan array berikutnya. **int j=i+1; j<listMhs.length; j++** digunakan untuk mengurutkan nilai yang belum terurut. **listMhs[j].ipk<listMhs[idxMin].ipk** digunakan untuk menentukan nilai apa yang akan menjadi perbandingan selanjutnya.

Percobaan 4

- Tambahkan kode pada MahasiswaBerprestasi07.java

```
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;
    }
}
```

- Tambahkan kode pada MahasiswaDemo07.java

```
System.out.println("Data yang sudah terurut menggunakan INSERTION SORT (ASC) ");
list.selectionSort();
list.tampil();
```

Hasil

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

```

-----
Data yang sudah terurut menggunakan INSERTION SORT (ASC)
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
-----
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>

```

Pertanyaan

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

```

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;
    }
}

```

Hasil

```

-----
Data yang sudah terurut menggunakan INSERTION SORT (ASC)
Nama   : ila
NIM    : 333
Kelas : 2c
IPK    : 3.8
-----
Nama   : ayu
NIM    : 111
Kelas : 2c
IPK    : 3.7
-----
Nama   : yayuk
NIM    : 555
Kelas : 2c
IPK    : 3.4
-----
Nama   : susi
NIM    : 444
Kelas : 2c
IPK    : 3.1
-----
Nama   : dika
NIM    : 222
Kelas : 2c
IPK    : 3.0
-----
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>

```

Latihan

Berdasarkan class diagram diatas buatlah menu dikelas main dengan pilihan menu:

1. Tambah data digunakan untuk menambahkan data dosen
2. Tampil data digunakan untuk menampilkan data seluruh dosen
3. Sorting ASC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari dosen termuda ke dosen tertua menggunakan bubble Sort.
4. Sorting DSC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari tertua ke dosen termuda dapat menggunakan algoritma selection sort atau insertion sort.

- Dosen07.java

```
package jobsheet_6;

public class Dosen07 {
    public String kode;
    public String nama;
    public boolean jenisKelamin;
    public int usia;
    public Dosen07() {
    }
    public Dosen07(String kd, String name, boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }
    public void tampilkanInformasi() {
        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("-----");
    }
}
```

- DataDosen07.java

```
package jobsheet_6;

public class DataDosen07 {
    Dosen07 [] listDosen = new Dosen07[10];
    int idx;
    void tambah (Dosen07 m) {
        if (idx<listDosen.length) {
            listDosen[idx]=m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }
    void tampil() {
        for (int i = 0; i < idx; i++) {
            Dosen07 m = listDosen[i];
            m.tampilkanInformasi();
            System.out.println("-----");
        }
    }
    void bubbleSort(){
        for (int i=0; i<idx-1; i++){
            for (int j=1; j<idx-i; j++){
                if (listDosen[j-1].usia>listDosen[j].usia) {
                    Dosen07 tmp = listDosen[j];
                    listDosen[j]=listDosen[j-1];
                    listDosen[j-1]=tmp;
                }
            }
        }
    }
    void selectionSort() {
        for (int i=0; i<idx-1; i++) {
            int idxMax=i;
            for (int j=i+1; j<idx; j++) {
                if(listDosen[j].usia>listDosen[idxMax].usia) {
                    idxMax=j;
                }
            }
            Dosen07 tmp = listDosen[idxMax];
            listDosen[idxMax]=listDosen[i];
            listDosen[i]=tmp;
        }
    }
    void insertionSort(){
        for (int i=1; i<idx; i++) {
            Dosen07 temp = listDosen[i];
            int j=i;
            while (j>0 && listDosen[j-1].usia<temp.usia) {
                listDosen[j]=listDosen[j-1];
                j--;
            }
            listDosen[j]=temp;
        }
    }
}
```

- DosenDemo07.java

```
package jobsheet_6;
import java.util.Scanner;
public class DosenDemo07 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen07 list = new DataDosen07();
        System.out.print("Masukkan jumlah dosen(1 - 10!) : ");
        int jmlDsn = sc.nextInt();
        sc.nextLine();
        System.out.println("-----
        -----");
        for (int i=0; i<jmlDsn; i++) {
            System.out.println("Masukkan Data Dosen ke-" + (i + 1));
            System.out.print("Kode          : ");
            String kode = sc.nextLine();
            System.out.print("Nama          : ");
            String nama = sc.nextLine();
            System.out.print("Jenis Kelamin : ");
            String jk = sc.nextLine();
            boolean jenisKelamin = jk.equalsIgnoreCase("Pria");
            System.out.print("Usia          : ");
            int usia = sc.nextInt();
            sc.nextLine();
            System.out.println("-----");
            Dosen07 dosen = new Dosen07(kode, nama, jenisKelamin, usia);
            list.tambah(dosen);
        }
        System.out.println("Data dosen sebelum sorting: ");
        list.tampil();

        System.out.println("Data dosen yang sudah terurut menggunakan
        BUBBLE SORT (ASC) : ");
        list.bubbleSort();
        list.tampil();

        System.out.println("Data dosen yang sudah terurut menggunakan
        SELECTION SORT (DESC)");
        list.selectionSort();
        list.tampil();

        System.out.println("Data dosen yang sudah terurut menggunakan
        INSERTION SORT (DESC)");
        list.insertionSort();
        list.tampil();
        sc.close();
    }
}
```


Hasil Latian

Masukkan jumlah dosen(1 - 10!) : 4

Masukkan Data Dosen ke-1

Kode : 111
Nama : Ayu
Jenis Kelamin : Wanita
Usia : 27

Masukkan Data Dosen ke-2

Kode : 222
Nama : Bima
Jenis Kelamin : Pria
Usia : 29

Masukkan Data Dosen ke-3

Kode : 333
Nama : Clara
Jenis Kelamin : Wanita
Usia : 48

Masukkan Data Dosen ke-4

Kode : 444
Nama : Dodik
Jenis Kelamin : Pria
Usia : 25

Data dosen sebelum sorting:

Kode : 111
Nama : Ayu
Jenis Kelamin : Perempuan
Usia : 27

Kode : 222
Nama : Bima
Jenis Kelamin : Laki-laki
Usia : 29

Kode : 333
Nama : Clara
Jenis Kelamin : Perempuan
Usia : 48

Kode : 444
Nama : Dodik
Jenis Kelamin : Laki-laki
Usia : 25

Data dosen yang sudah terurut menggunakan BUBBLE SORT (ASC) :

Kode : 444
Nama : Dodik
Jenis Kelamin : Laki-laki
Usia : 25

Kode : 111
Nama : Ayu
Jenis Kelamin : Perempuan
Usia : 27

Kode : 222
Nama : Bima
Jenis Kelamin : Laki-laki
Usia : 29

Kode : 333
Nama : Clara
Jenis Kelamin : Perempuan
Usia : 48

```
-----  
Data dosen yang sudah terurut menggunakan SELECTION SORT (DESC)  
Kode      : 333  
Nama      : Clara  
Jenis Kelamin : Perempuan  
Usia      : 48  
-----
```

```
-----  
Kode      : 222  
Nama      : Bima  
Jenis Kelamin : Laki-laki  
Usia      : 29  
-----
```

```
-----  
Kode      : 111  
Nama      : Ayu  
Jenis Kelamin : Perempuan  
Usia      : 27  
-----
```

```
-----  
Kode      : 444  
Nama      : Dodik  
Jenis Kelamin : Laki-laki  
Usia      : 25  
-----
```

```
-----  
Data dosen yang sudah terurut menggunakan INSERTION SORT (DESC)  
Kode      : 333  
Nama      : Clara  
Jenis Kelamin : Perempuan  
Usia      : 48  
-----
```

```
-----  
Kode      : 222  
Nama      : Bima  
Jenis Kelamin : Laki-laki  
Usia      : 29  
-----
```

```
-----  
Kode      : 111  
Nama      : Ayu  
Jenis Kelamin : Perempuan  
Usia      : 27  
-----
```

```
-----  
Kode      : 444  
Nama      : Dodik  
Jenis Kelamin : Laki-laki  
Usia      : 25  
-----
```

```
-----  
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>
```