

# **Laporan Hasil Praktikum Algoritma Struktur Data**

## **Jobsheet 6**



**Febryan Akhmad Taajuddin**

**244107020180**

**Kelas 1E**

**Program Studi Teknik Informatika**

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

**2025**

## Percobaan 1 (A) SORTING – BUBBLE SORT

1. Buat class Sorting10 lalu tambahkan atribut

```
public class Sorting10 {  
    int [] data;  
    int jumData;  
  
}
```

2. Buatlah konstruktor dengan parameter Dara[] dan jmlDat

```
Sorting10 (int Data[], int jmlDat){  
    jumData = jmlDat;  
    data = new int [jmlDat];  
    for (int i = 0; i < jumData; i++) {  
        data[i] = Data [i];  
    }  
}
```

3. Buatlah method bubblesort bertipe void

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

4. Buatlah method tampil bertipe void

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

5. Buatlah class SortingMain kemudian deklarasi array dengan nama a[]

```
public class SortingMain10 {  
    public static void main(String[] args) {  
        int a[] = {20, 10, 2, 7, 12};  
    }  
}
```

6. Buatlah objek baru dengan nama dataurut1

```
Sorting10 dataurut1 = new Sorting10(a, a.length);
```

7. Lakukan pemanggilan method bubblesort dan tampil

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

8. Run kode program

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

## Percobaan 1 (B) SORTING – SELECTION SORT

1. Pada class Sorting10 tambahkan methos SelectionSort

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

2. Buat deklarasi array pada kelas SortingMain10

```
int b[] = {30, 20, 2, 8, 14};
```

3. Buat objek baru dengan nama dataurut2

```
Sorting10 dataurut2 = new Sorting10(b, b.length);
```

4. Lakukan pemanggilan method SelectionSort dan tampil

```
System.out.println("Data awal 2");
dataurut2.tampil();
dataurut2.SelectionSort();
System.out.println("Data sudah diurutkan dengan
SELECTION SORT (ASC)");
dataurut2.tampil();
```

5. Run kode program

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

## Percobaan 1 (C) SORTING – INSERTION SORT

1. Pada class Sorting10 tambahkan methos insertionSort

```
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. Buat deklarasi array pada kelas SortingMain10

```
int c[] = {40, 10, 4, 9, 3};
```

3. Buat objek baru dengan nama dataurut3

```
Sorting10 dataurut3 = new Sorting10(c, c.length);
```

4. Lakukan pemanggilan method insertionSort dan tampil

```
System.out.println("Data awal 3");
dataurut3.tampil();
dataurut3.insertionSort();
System.out.println("Data sudah diurutkan dengan
INSERTION SORT (ASC)");
dataurut3.tampil();
```

5. Run kode program

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

## Pertanyaan percobaan 1

1. Jelaskan fungsi kode program berikut

```
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 percobaan 1

1. Jika data indeks array sebelumnya lebih besar dari data indeks sekarang maka indeks sekarang disimpan dulu ke variable temp, lalu indeks sekarang adalah data sebelumnya (digeser ke kanan), lalu data sebelumnya diambil dari variable temp.
2. Berikut kode programnya

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

3. Perulangan akan terus terjadi saat j masih dalam batas array  $j \geq 0$ , j lebih besar dari temp yang sedang di proses. Hal ini bertujuan perulangan digunakan untuk menggeser elemen yang lebih besar dari temp ke kanan, sehingga temp bisa dimasukkan ke kiri.
4. Kode digunakan untuk menggeser elemen temp yang lebih besar ke kanan, sehingga elemen yang lebih kecil bisa dimasukkan ke kiri.

## Percobaan 2 SORTING IPK – BUBBLE SORT

### 1. Buat class Mahasiswa10

```
public class Mahasiswa10 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa10(){
    }

    Mahasiswa10 (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);
    }
}
```

### 2. Buat class MahasiswaBerprestasi10

```
public class MahasiswaBerprestasi10 {
    Mahasiswa10 [] listMhs = new Mahasiswa10[5];
    int idx;
}
```

### 3. Tambahkan method tambah()

```
void tambah(Mahasiswa10 m){
    if (idx < listMhs.length) {
        listMhs[idx] = m;
        idx++;
    } else {
        System.err.println("data sudah penuh");
    }
}
```

### 4. Tambahkan method tampil()

```
void tampil(){
    for (Mahasiswa10 m:listMhs) {
        m.tampilInformasi();
        System.out.println("-----");
    }
}
```

## 5. Tambahkan method bubble sort

```
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) {
                Mahasiswa10 tmp = listMhs[j];
                listMhs[j]= listMhs[j-1];
                listMhs[j-1]=tmp;
            }
        }
    }
}
```

## 6. Buat class MahasiswaDemo10

```
public class MahasiswaDemo10 {
    public static void main(String[] args) {
        MahasiswaBerprestasi10 list = new MahasiswaBerprestasi10();
        Mahasiswa10 m1 = new Mahasiswa10("123", "Zidan", "2A", 3.2);
        Mahasiswa10 m2 = new Mahasiswa10("124", "Ayu", "2A", 3.5);
        Mahasiswa10 m3 = new Mahasiswa10("125", "Sofi", "2A", 3.1);
        Mahasiswa10 m4 = new Mahasiswa10("126", "Sita", "2A", 3.9);
        Mahasiswa10 m5 = new Mahasiswa10("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();
    }
}
```

7. Run kode program

```
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 setelah 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
-----
IPK: 3.2
Kelas: 2A
-----
Nama: Sofi
NIM: 125
IPK: 3.1
Kelas: 2A
-----
```



## 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. Jawaban
  - a. Untuk membatasi jumlah iterasi agar algoritma berhenti setelah seluruh array terurut
  - b. Untuk meningkatkan efisiensi algoritma dengan mengurangi jumlah iterasi pada loop dalam j seiring dengan bertambahnya iterasi loop luar i
  - c. Perulangan i akan berlangsung sebanyak 49 kali dan tahap bubble sort yang ditempuh sebanyak 49 kali
2. Modifikasi kode program hanya pada class MahasiswaDemo10, berikut kodenya

```
import java.util.Scanner;  
public class MahasiswaDemo10 {  
    public static void main(String[] args) {  
        MahasiswaBerprestasi10 list = new MahasiswaBerprestasi10();  
        Scanner sc = new Scanner(System.in);  
        for (int i = 0; i < 5; i++) {  
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));  
            System.out.print("Nama: ");  
            String nama = sc.nextLine();  
            System.out.print("NIM: ");  
            String nim = sc.nextLine();  
            System.out.print("IPK: ");  
            double ipk = sc.nextDouble();  
            sc.nextLine();  
            System.out.print("Kelas: ");  
            String kelas = sc.nextLine();  
  
            list.tambah(new Mahasiswa10(nim, nama, kelas, ipk));  
        }  
        System.out.println("Data mahasiswa sebelum sorting: ");  
        list.tampil();  
  
        System.out.println("Data Mahasiswa setelah sorting berdasarkan  
IPK (DESC): ");  
        list.bubbleSort();  
        list.tampil();  
    }  
}
```

- Output

```
Masukkan data mahasiswa ke-1
Nama: agus
NIM: 123
IPK: 3,2
Kelas: 2a
Masukkan data mahasiswa ke-2
Nama: munas
NIM: 124
IPK: 3,5
Kelas: 2a
Masukkan data mahasiswa ke-3
Nama: hardi
NIM: 125
IPK: 3,1
Kelas: 2a
Masukkan data mahasiswa ke-4
Nama: derajat
NIM: 126
IPK: 3,9
Kelas: 2a
Masukkan data mahasiswa ke-5
Nama: subkhan
NIM: 127
IPK: 3,7
Kelas: 2a
```

```
Data mahasiswa sebelum sorting:
Nama: agus
NIM: 123
IPK: 3.2
Kelas: 2a
-----
Nama: munas
NIM: 124
IPK: 3.5
Kelas: 2a
-----
Nama: hardi
NIM: 125
IPK: 3.1
Kelas: 2a
-----
Nama: derajat
NIM: 126
IPK: 3.9
Kelas: 2a
-----
Nama: subkhan
NIM: 127
IPK: 3.7
Kelas: 2a
-----
```

```

Data Mahasiswa setelah sorting berdasarkan IPK (DESC):
Nama: derajat
NIM: 126
IPK: 3.9
Kelas: 2a
-----
Nama: subkhan
NIM: 127
IPK: 3.7
Kelas: 2a
-----
Nama: munas
NIM: 124
IPK: 3.5
Kelas: 2a
-----
Nama: agus
NIM: 123
IPK: 3.2
Kelas: 2a
-----
Nama: hardi
NIM: 125
IPK: 3.1
Kelas: 2a
-----

```

## Percobaan 2 SORTING IPK – SELECTION SORT

1. Tambahkan method selectionSort() pada class MahasiswaBerprestasi10

```

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;
            }
        }
        Mahasiswa10 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs [i];
        listMhs[i] = tmp;
    }
}

```

2. Tambahkan method tampil pada class MahasiswaDemo10

```

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

```

3. Run kode program

```
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)
Nama: ila
NIM: 124
IPK: 3.1
Kelas: 2B
-----
Nama: udin
NIM: 127
IPK: 3.2
Kelas: 2B
-----
Nama: tika
NIM: 126
IPK: 3.3
Kelas: 2B
-----
Nama: agus
NIM: 125
IPK: 3.6
Kelas: 2B
-----
Nama: Ali
NIM: 123
IPK: 3.9
Kelas: 2B
-----
```

## Pertanyaan

1. 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. Program tersebut bertujuan untuk mengurutkan data mahasiswa berdasarkan IPK secara ascending dari kecil ke besar. int idxMin = i adalah indeks elemen dengan nilai IPK terkecil, lalu dilakukan perulangan untuk mencari elemen dengan nilai IPK terkecil sesuai panjang length array, lalu jika indeks j memiliki IPK lebih kecil dari indeks idxMin maka idxMin menjadi j.

## Percobaan 2 SORTING IPK – INSERTION SORT

1. Tambahkan method insertionSort() pada class MahasiswaBerprestasi10

```
void insertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa10 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. Tambahkan method tampil pada class MahasiswaDemo10

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

3. Run kode program

```
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)

```
Nama: dika
NIM: 222
IPK: 3.0
Kelas: 2c
```

-----

```
Nama: susi
NIM: 444
IPK: 3.1
Kelas: 2c
```

-----

```
Nama: yayuk
NIM: 555
IPK: 3.4
Kelas: 2c
```

-----

```
Nama: ayu
NIM: 111
IPK: 3.7
Kelas: 2c
```

-----

```
Nama: ila
NIM: 333
IPK: 3.8
Kelas: 2c
```

-----

## Pertanyaan

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

## Jawaban

1. Kode program

```
void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa10 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j - 1].ipk < temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

- Output

```
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)
Nama: ila
NIM: 333
IPK: 3.8
Kelas: 2c
-----
Nama: ayu
NIM: 111
IPK: 3.7
Kelas: 2c
-----
Nama: yayuk
NIM: 555
IPK: 3.4
Kelas: 2c
-----
Nama: susi
NIM: 444
IPK: 3.1
Kelas: 2c
-----
Nama: dika
NIM: 222
IPK: 3.0
Kelas: 2c
-----
```

## Latihan Praktikum

### 1. Berikut kode programnya

- Class Dosen10

```
public class Dosen10 {
    String kode;
    String nama;
    String jenisKelamin;
    int usia;

    Dosen10() {

    }
    Dosen10(String kd, String name, String jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }
    void tampil() {
        System.out.println("Kode: " + kode);
        System.out.println("Nama: " + nama);
        System.out.println("Jenis kelamin: " + jenisKelamin);
        System.out.println("Usia: " + usia);
    }
}
```



- Class DataDosen10

```
public class DataDosen10 {
    Dosen10 [] dataDosen = new Dosen10[10];
    int idx;

    void tambah(Dosen10 m){
        if (idx < dataDosen.length) {
            dataDosen[idx] = m;
            idx++;
        } else {
            System.err.println("data sudah penuh");
        }
    }
    void tampil(){
        for (Dosen10 m:dataDosen) {
            m.tampil();
            System.out.println("-----");
        }
    }
    void bubbleSort(){
        for (int i = 0; i < dataDosen.length-1; i++) {
            for (int j = 1; j < dataDosen.length-i; j++) {
                if (dataDosen[j].usia<dataDosen[j-1].usia) {
                    Dosen10 tmp = dataDosen[j];
                    dataDosen[j]= dataDosen[j-1];
                    dataDosen[j-1]=tmp;
                }
            }
        }
    }
    void insertionSort() {
        for (int i = 1; i < dataDosen.length; i++) {
            Dosen10 temp = dataDosen[i];
            int j = i;
            while (j>0 && dataDosen[j - 1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j - 1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}
```

- Class DosenMain10

```
import java.util.Scanner;
public class DosenMain10 {
    public static void main(String[] args) {
        DataDosen10 list = new DataDosen10();
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 10; 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 (L/P): ");
            String jenisKelamin = sc.nextLine();
            System.out.print("Usia: ");
            int usia = sc.nextInt();
            sc.nextLine();
            System.out.println("-----");

            list.tambah(new Dosen10(kode, nama, jenisKelamin, usia));
        }
        System.out.println("Data dosen sebelum sorting: ");
        list.tampil();

        System.out.println("Data dosen setelah sorting menggunakan
BUBBLE SORT (ASC): ");
        list.bubbleSort();
        list.tampil();

        System.out.println("Data dosen setelah sorting menggunakan
INSERTION SORT (DSC)");
        list.insertionSort();
        list.tampil();
    }
}
```

- Output

```
Data dosen sebelum sorting:
```

```
Kode: 1
```

```
Nama: Candra
```

```
Jenis kelamin: L
```

```
Usia: 32
```

```
-----
```

```
Kode: 2
```

```
Nama: Agus
```

```
Jenis kelamin: L
```

```
Usia: 30
```

```
-----
```

```
Kode: 3
```

```
Nama: Huda
```

```
Jenis kelamin: L
```

```
Usia: 29
```

```
-----
```

```
Kode: 4
```

```
Nama: Munas
```

```
Jenis kelamin: L
```

```
Usia: 22
```

```
-----
```

```
Kode: 4
```

```
Nama: Teguh
```

```
Jenis kelamin: L
```

```
Usia: 35
```

```
-----
```

```
Kode: 6
```

```
Nama: Roni
```

```
Jenis kelamin: L
```

```
Usia: 50
```

```
-----
```

```
Kode: 7
```

```
Nama: Dirman
```

```
Jenis kelamin: L
```

```
Usia: 34
```

```
-----
```

```
Kode: 8
```

```
Nama: Derajat
```

```
Jenis kelamin: L
```

```
Usia: 37
```

```
-----
```

```
Kode: 9
```

```
Nama: Subkhan
```

```
Jenis kelamin: L
```

```
Usia: 25
```

```
-----
```

```
Kode: 10
```

```
Nama: Budi
```

```
Jenis kelamin: L
```

```
Usia: 21
```

```
-----
```

Data dosen setelah sorting menggunakan BUBBLE SORT (ASC):

Kode: 10

Nama: Budi

Jenis kelamin: L

Usia: 21

-----

Kode: 4

Nama: Munas

Jenis kelamin: L

Usia: 22

-----

Kode: 9

Nama: Subkhan

Jenis kelamin: L

Usia: 25

-----

Kode: 3

Nama: Huda

Jenis kelamin: L

Usia: 29

-----

Kode: 2

Nama: Agus

Jenis kelamin: L

Usia: 30

-----

Kode: 1

Nama: Candra

Jenis kelamin: L

Usia: 32

-----

Kode: 7

Nama: Dirman

Jenis kelamin: L

Usia: 34

-----

Kode: 4

Nama: Teguh

Jenis kelamin: L

Usia: 35

-----

Kode: 8

Nama: Derajat

Jenis kelamin: L

Usia: 37

-----

Kode: 6

Nama: Roni

Jenis kelamin: L

Usia: 50

-----

Data dosen setelah sorting menggunakan INSERTION SORT (DSC)

Kode: 6

Nama: Roni

Jenis kelamin: L

Usia: 50

-----  
Kode: 8

Nama: Derajat

Jenis kelamin: L

Usia: 37

-----  
Kode: 4

Nama: Teguh

Jenis kelamin: L

Usia: 35

-----  
Kode: 7

Nama: Dirman

Jenis kelamin: L

Usia: 34

-----  
Kode: 1

Nama: Candra

Jenis kelamin: L

Usia: 32

-----  
Kode: 2

Nama: Agus

Jenis kelamin: L

Usia: 30

-----  
Kode: 3

Nama: Huda

Jenis kelamin: L

Usia: 29

-----  
Kode: 9

Nama: Subkhan

Jenis kelamin: L

Usia: 25

-----  
Kode: 4

Jenis kelamin: L

Usia: 22

-----  
Kode: 10

Nama: Budi

Jenis kelamin: L

Usia: 21

-----