

POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA



Nama	: Muhammad Nuril Huda
Kelas	: TI-1A
No	: 19
Mata Kuliah	: Algoritma dan Struktur Data

5.2 Praktikum 1- Mengimplementasikan Sorting menggunakan object

5.2.1 Kode Program

➤ Kode Program Sorting19

```
package Pertemuan6;

public class Sorting19 {

    int [] data;

    int jumData;

    Sorting19 (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 tampil(){

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

            System.out.print(data[i]+" ");

        }

        System.out.println();

    }

}
```

```

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;
    }
}
}

```

➤ Kode Program SortingMain19

```

package Pertemuan6;

public class SortingMain19 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        Sorting19 dataurut1 = new Sorting19(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};

        Sorting19 dataurut2 = new Sorting19(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};

        Sorting19 dataurut3 = new Sorting19(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();

    }
}

```

5.2.2 Hasil Percobaan

```

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

```

5.2.3 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;
}

```

- Maksud dari kode program tersebut adalah menukar nilai data[j] dengan data[j-1] menggunakan bantuan variable temp untuk menyimpan data sementara.

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

```
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;  
}
```

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

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

- Kondisi ini digunakan untuk menggeser elemen yang lebih besar dari temp ke kanan agar temp dapat dimasukkan ke posisi yang benar. $j \geq 0$ memastikan bahwa perulangan tidak keluar dari batas kiri array, sementara $\text{data}[j] > \text{temp}$ memastikan elemen yang lebih besar dari temp digeser. Proses ini berlanjut hingga ditemukan posisi yang tepat untuk menempatkan temp, sehingga array tetap dalam keadaan terurut selama iterasi.

4. Pada Insertion sort, apakah tujuan dari perintah

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

- Perintah ini digunakan untuk menggeser angka ke kanan agar ada tempat bagi angka yang akan dimasukkan. Jika angka sebelumnya lebih besar dari angka yang sedang diproses, angka tersebut dipindahkan ke kanan. Proses ini terus dilakukan sampai ditemukan posisi yang tepat untuk angka yang akan disisipkan.

5.3 Praktikum 2- (Sorting Menggunakan Array of Object)

5.3.1 Kode Program (Bubble Sort)

➤ Kode Program Mahasiswa19

```
package Pertemuan6;

public class Mahasiswa19 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    // Konstruktor default

    Mahasiswa19 () {

    }

    // Konstruktor berparameter (dibuat ada yang nama var parameter inputnya
    sama ada yang tidak)

    Mahasiswa19(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("Kelas: " + kelas);

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

    }

}
```

➤ Kode Program MahasiswaBerprestasi19

```
package Pertemuan6;

public class MahasiswaBerprestasi19 {
    Mahasiswa19 [] listMhs = new Mahasiswa19[5];
    int idx;
    void tambah (Mahasiswa19 m){
        if (idx<listMhs.length){
            listMhs[idx]=m;
            idx++;
        }else {
            System.out.println("data sudah penuh");
        }
    }
    void tampil (){
        for (Mahasiswa19 m : listMhs){
            m.tampilInformasi();
            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){
                    Mahasiswa19 tmp = listMhs[j];
                    listMhs[j]=listMhs[j-1];
                    listMhs[j-1]=tmp;
                }
            }
        }
    }
}
```

➤ Kode Program MahasiswaDemo19

```
package Pertemuan6;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        MahasiswaBerprestasi19 list = new MahasiswaBerprestasi19();

        Mahasiswa19 m1 = new Mahasiswa19 ("123", "Zidan", "2A",3.2);
        Mahasiswa19 m2 = new Mahasiswa19 ("124", "Ayu", "2A",3.5);
        Mahasiswa19 m3 = new Mahasiswa19 ("125", "Sofi", "2A",3.1);
        Mahasiswa19 m4 = new Mahasiswa19 ("126", "Sita", "2A",3.9);
        Mahasiswa19 m5 = new Mahasiswa19 ("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();
    }
}
```


5.3.2 Hasil Percobaan (Bubble Sort)

```
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
-----
```

5.3.3 Pertanyaan (Bubble Sort)

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 $i < \text{listMhs.length} - 1$?
 - Karena setiap Langkah BubbleSort descending akan menempatkan mahasiswa dengan IPK terbesar ke posisi indeks paling kiri. Setelah $\text{listMhs.length} - 1$ langkah, semua mahasiswa sudah terurut berdasarkan IPK, sehingga tidak perlu melakukan perulangan sebanyak listMhs.length .
 - b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - Karena setiap kali perulangan i berjalan, mahasiswa dengan IPK tertinggi akan berada diposisi yang sudah benar. Oleh karena itu, jumlah perbandingan berkurang setiap kali i bertambah agar tidak memeriksa IPK yang sudah benar posisinya.
 - c. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?
 - Perulangan i akan berlangsung sebanyak 49 kali, karena $i < 50 - 1$.
 - Bubble sort akan menempuh 49 tahap, karena setiap tahap menempatkan satu mahasiswa dengan IPK tertinggi ke posisi akhirnya hingga seluruh daftar mahasiswa terurut.
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

```
package Pertemuan6;  
  
public class MahasiswaBerprestasi19 {  
    Mahasiswa19[] listMhs;  
  
    int idx;  
  
    MahasiswaBerprestasi19(int jumlah) {  
        listMhs = new Mahasiswa19[jumlah];  
        idx = 0;  
    }  
}
```

```

package Pertemuan6;

import java.util.Scanner;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan jumlah mahasiswa: ");

        int jumlah = sc.nextInt();

        sc.nextLine();

        MahasiswaBerprestasi19 list = new
MahasiswaBerprestasi19(jumlah);

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

            System.out.println("\nMasukkan data mahasiswa ke-" + (i +
1));

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

            String nama = sc.nextLine();

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

            String nim = sc.nextLine();

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

            String kelas = sc.nextLine();

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

            double ipk = sc.nextDouble();

            sc.nextLine();

            list.tambah(new Mahasiswa19(nama, nim, 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();

    }

}

```

5.3.4 Kode Program (Selection Sort)

➤ Kode Program MahasiswaBerprestasi19

```
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;  
            }  
        }  
        Mahasiswa19 tmp = listMhs[idxMin];  
        listMhs[idxMin]=listMhs[i];  
        listMhs[i]=tmp;  
    }  
}
```

➤ Kode Program MahasiswaDemo19

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

5.3.5 Hasil Percobaan (Selection Sort)

```
Data yang sudah terurut menggunakan SELECTION SORT (ASC)  
Nama: 123  
NIM: Nuril  
Kelas: 3  
IPK: 3.0  
-----  
Nama: 345  
NIM: Huda  
Kelas: 4  
IPK: 4.0  
-----
```

5.3.6 Pertanyaan (Selection Sort)

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!

- Kode tersebut digunakan untuk mencari mahasiswa dengan IPK terkecil dalam daftar yang belum diurutkan. Pertama, indeks awal (idxMin) diset sebagai nilai terkecil sementara. Kemudian, perulangan dilakukan untuk membandingkan IPK mahasiswa lainnya yang berada setelah posisi tersebut. Jika ditemukan mahasiswa dengan IPK lebih kecil, maka indeksnya disimpan ke dalam idxMin. Setelah proses pencarian selesai, idxMin akan berisi indeks mahasiswa dengan IPK terkecil, yang nantinya akan ditukar dengan posisi awal. Proses ini dilakukan berulang hingga seluruh data tersusun dari IPK terkecil hingga terbesar.

5.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort Waktu

5.4.1 Kode Program

➤ Kode Program MahasiswaBerprestasi19

```
void insertionSort(){
    for (int i=1; i<listMhs.length; i++){
        Mahasiswa19 temp = listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk){
            listMhs[j]=listMhs[j-1];
            j--;
        }
        listMhs[j]=temp;
    }
}
```

➤ Kode Program MahasiswaDemo19

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

5.4.2 Hasil Percobaan

```
Data yang sudah terurut menggunakan INSERTION SORT (ASC)
Nama: 575756765
NIM: Huda
Kelas: 2
IPK: 3.5
-----
Nama: 12312312
NIM: Nuril
Kelas: 1
IPK: 4.0
```

5.4.3 Pertanyaan

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

```
void insertionSort(){
    for (int i = 1; i < listMhs.length; i++){
        Mahasiswa19 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 yang sudah terurut menggunakan INSERTION SORT (DESC)
Nama: 123123
NIM: Nuril
Kelas: 2
IPK: 4.0
-----
Nama: 3241324
NIM: Huda
Kelas: 1
IPK: 2.0
-----
```

5.5 Latihan Praktikum

5.5.1 Kode Program

➤ Kode Program Dosen19

```
package Pertemuan6;

public class Dosen19 {

    String kode, nama;

    Boolean jenisKelamin;

    int usia;

    Dosen19(String kd, String name, Boolean jk, int age){

        kode = kd;

        nama = name;

        jenisKelamin = jk;

        usia = age;

    }

    void tampil(){

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

        System.out.println("Kode\t: "+kode);

        System.out.println("Jenis Kelamin\t: "+(jenisKelamin ? "Laki-laki" : "Perempuan"));

        System.out.println("Usia\t: "+usia);

    }

}
```

➤ Kode Program DataDosen19

```
package Pertemuan6;

public class DataDosen {

    Dosen19 [] dataDosen = new Dosen19 [10];

    int idx;

    void tambah (Dosen19 m){

        if (idx<dataDosen.length){

            dataDosen[idx]=m;

            idx++;

        }else {

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

        }

    }

}
```

```

void tampil (){
    for (int i = 0; i < idx; i++) {
        dataDosen[i].tampil();
        System.out.println("-----");
    }
}

void bubbleSort(){
    for (int i = 0; i < idx - 1; i++){
        for (int j = 1; j < idx - i; j++){
            if (dataDosen[j] != null && dataDosen[j - 1] != null) {
                if (dataDosen[j].usia < dataDosen[j - 1].usia){
                    Dosen19 tmp = dataDosen[j];
                    dataDosen[j] = dataDosen[j - 1];
                    dataDosen[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 (dataDosen[j] != null && dataDosen[idxMax] != null) {
                if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                    idxMax = j;
                }
            }
        }
        if (idxMax != i) {
            Dosen19 tmp = dataDosen[idxMax];
            dataDosen[idxMax] = dataDosen[i];
            dataDosen[i] = tmp;
        }
    }
}

```



```

void insertionSort() {
    for (int i = 1; i < idx; i++) {
        Dosen19 temp = dataDosen[i];
        int j = i;
        if (temp != null) {
            while (j > 0 && dataDosen[j - 1] != null && dataDosen[j - 1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j - 1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}
}

```

➤ Kode Program DosenMain19

```

package Pertemuan6;

import java.util.Scanner;

public class DosenMain19 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen data = new DataDosen();
        while (true) {
            System.out.println("\n===== MENU DATA DOSEN =====");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampilkan Data Dosen Sebelum Disorting");
            System.out.println("3. Sorting (Bubble Sort - Ascending Usia)");
            System.out.println("4. Sorting (Selection Sort - Descending
Usia)");
            System.out.println("5. Sorting (Insertion Sort - Decending Usia)");
            System.out.println("6. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

```

```

        if (pilihan == 1) {
            System.out.println("\nMasukkan Data Dosen");
            System.out.print("Kode Dosen: ");
            String kode = sc.nextLine();
            System.out.print("Nama Dosen: ");
            String nama = sc.nextLine();
            System.out.print("Jenis Kelamin (L/P): ");
            String kelamin = sc.nextLine();
            boolean jenisKelamin = kelamin.equalsIgnoreCase("L");
            System.out.print("Usia Dosen: ");
            int usia = sc.nextInt();
            sc.nextLine();

            data.tambah(new Dosen19(kode, nama, jenisKelamin, usia));
            System.out.println("Data dosen berhasil ditambahkan!");

        } else if (pilihan == 2) {
            System.out.println("\n===== DATA DOSEN SEBELUM DISORTING =====");
            data.tampil();
        } else if (pilihan == 3) {
            System.out.println("\nSorting menggunakan Bubble Sort ");
            data.bubbleSort();
            data.tampil();
        } else if (pilihan == 4) {
            System.out.println("\nSorting menggunakan Selection Sort ");
            data.selectionSort();
            data.tampil();
        } else if (pilihan == 5) {
            System.out.println("\nSorting menggunakan Insertion Sort ");
            data.insertionSort();
            data.tampil();
        } else if (pilihan == 6) {
            System.out.println("Program Selesai. Terima Kasih");
            break;
        } else {
            System.out.println("Pilihan tidak valid! Silakan pilih kembali.");
        }
    }
}

```

5.5.2 Hasil

```
===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar
Pilih menu: 1

Masukkan Data Dosen
Kode Dosen: 1
Nama Dosen: Nuril
Jenis Kelamin (L/P): L
Usia Dosen: 23
Data dosen berhasil ditambahkan!

===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar
Pilih menu: 1

Masukkan Data Dosen
Kode Dosen: 2
Nama Dosen: Aurora
Jenis Kelamin (L/P): P
Usia Dosen: 45
Data dosen berhasil ditambahkan!

===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar
Pilih menu: 2

===== DATA DOSEN SEBELUM DISORTING =====
Nama   : Nuril
Kode    : 1
Jenis Kelamin : Laki-laki
Usia    : 23
-----
Nama   : Aurora
Kode    : 2
Jenis Kelamin : Perempuan
Usia    : 45
-----

===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar
Pilih menu: 3
```

Sorting menggunakan Bubble Sort

Nama : Nuril

Kode : 1

Jenis Kelamin : Laki-laki

Usia : 23

Nama : Aurora

Kode : 2

Jenis Kelamin : Perempuan

Usia : 45

===== MENU DATA DOSEN =====

1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar

Pilih menu: 4

Sorting menggunakan Selection Sort

Nama : Aurora

Kode : 2

Jenis Kelamin : Perempuan

Usia : 45

Nama : Nuril

Kode : 1

Jenis Kelamin : Laki-laki

Usia : 23

===== MENU DATA DOSEN =====

1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar

Pilih menu: 5

Sorting menggunakan Insertion Sort

Nama : Aurora

Kode : 2

Jenis Kelamin : Perempuan

Usia : 45

Nama : Nuril

Kode : 1

Jenis Kelamin : Laki-laki

Usia : 23

```
===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar
Pilih menu: 5

Sorting menggunakan Insertion Sort
Nama      : Aurora
Kode      : 2
Jenis Kelamin : Perempuan
Usia      : 45
-----
Nama      : Nuril
Kode      : 1
Jenis Kelamin : Laki-laki
Usia      : 23
-----

===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Keluar
Pilih menu: 6
Program Selesai. Terima Kasih
```

Link Github : <https://github.com/nurilhuda05/Algoritma-dan-Struktur-Data/tree/0b2301d01128cf02bed23f9554feb2fddb13614b/Pertemuan6>