Laporan Hasil Praktikum
Algoritma dan Struktur Data
Jobsheet 6
SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

Muhammad Firman Aditiasmara

244107020094

TI-1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri malang

2025

1. **Tujuan Praktikum**

   Setelah melakukan materi praktikum ini, mahasiswa mampu:

   a. Mahasiswa mampu membuat algoritma sorting menggunakan bubble sort, selection sort dan insertion sort
   b. Mahasiswa mampu menerapkan algoritma sorting menggunakan bubble sort, selection sort dan insertion sort pada program

2. **Praktikum**

   2.1 Percobaan 1: Mengimplementasikan Sorting menggunakan object

   2.1.1 Langkah-langkah Percobaan

   a. SORTING – BUBBLE SORT

   1. Membuat class Sorting<Nomor Presensi>,melengkapi dengan atribut dan method dan membuat konstruktor dengan parameter Data[] dan jmlDat

```
public class Sorting18 {

    int[] data;
    int jumData;

    Sorting18(int Data[], int jmlDat) {
        jumData = jmlDat;
        data = new int[jmlDat];
        for (int i = 0; i < jumData; i++) {
            data[i] = Data[i];
        }
    }
```

   2. Membuat methode bubbleSiort() dan tampil() bertipe void

```
  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();
     }
}
```

3. Membuat clas SortingMain<No presensi>, mendeklarasikan array a[] ,membuat objek baru dengan nama dataurut1 dan melakukan

```
public class SortingMain18 {

    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        Sorting18 dataurut1 = new Sorting18(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};
        Sorting18 dataurut2 = new Sorting18(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};
        Sorting18 dataurut3 = new Sorting18(c,
c.length);
        System.out.println("Data Awal 3");
        dataurut3.tampil();
        dataurut3.InstertionSort();
        System.out.println("Data sudah diurutkan dengan
INSERTION SORT (ASC)");
        dataurut3.tampil();

    }
}
```

pemanggilan method **bubbleSort** dan **tampil**

2.1.2 Verifikasi Hasil Percobaan

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

Hasil percobaan telah sesuai dengan ketentuan

b. **SORTING – SELECTION SORT**

1. Menambahkan method **SelectionSort** pada class Sorting

```
void SelectionSort() {
        for (int i = 0; i < jumData - 1; 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. Mendeklarasikan array b[] pada kelas **SortingMain**<No Presensi>, membuat objek dengan nama **dataurut2**, dan melakukan pemanggulan method **SelectionSort** dan **tampil**

```
 int b[] = {30, 20, 2, 8, 14};
        Sorting18 dataurut2 = new Sorting18(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();
```

2.1.3 Verifikasi Hasil Percobaan

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

Hasil Percobaan telah sesuai dengan ketentuan

c. **SORTING – INSERTION SORT**

1. Menambahkan method insertionSort pada kelas **Sorting**<No Presensi>

```
 void InstertionSort() {
        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. Mendeklarasikan array dengan nama c[] pada kelas **SortingMain**<No Presensi>, membuat objek dengan nama **dataurut3,** dan melakukan pemanggilan method **insertionSort** dan **tampil**

```
 int c[] = {40, 10, 4, 9, 3};
        Sorting18 dataurut3 = new Sorting18(c,
c.length);
        System.out.println("Data Awal 3");
        dataurut3.tampil();
        dataurut3.InstertionSort();
        System.out.println("Data sudah diurutkan
dengan INSERTION SORT (ASC)");
        dataurut3.tampil();
```

2.1.4 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 program tersebut merupakan kode program bubble sort Kode ini menukar p**osisi** dua elemen dalam array jika elemen sebelumnya (data[j-1]) lebih besar dari elemen saat ini (data[j]).

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

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

Jawab : pada kode tersebut int min = i; menyimpan nilai terkecil yang ditemukan, Ketika perulangan dilanjutkan, jika menemukan nilai minimum lagi maka min =j; akan diperbarui

3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan `while (j>=0 && data[j]>temp)`

Jawab : kode tersebut berfungsi untuk memastikan bahwa elemen-elemen yang lebih besar dari elemen yang sedang dimasukkan akan

digeser ke kanan hingga menemukan posisi yang tepat untuk menyisipkan elemen tersebut

4. Pada Insertion sort, apakah tujuan dari perintah `data [j+1] = data[j]`
   Jawaban : menggeser elemen yang lebih besar ke kanan agar dapat memasukkan elemen baru yang sedang dimasukkan.

2.2 Percobaan 2: Sorting Menggunakan Array of Object

2.2.1 Langkah-langkah Percobaan - Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

1. Membuat class **Mahasiswa**<No Presensi> dan Mengisi class Ttersebut dengan kode sesuai praktikum

```
public class Mahasiswa18 {

    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa18() {

    }

    Mahasiswa18(String nm, String name, String kls,
double ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }

    void tampilInformasi() {
        System.out.println("Nama    : " + nama);
        System.out.println("NIM     : " + nim);
        System.out.println("Kelas   : " + kelas);
        System.out.println("IPK     : " + ipk);
    }
}
```

2. Membuat class **MahasiswaBerprestasi**<No Presensi>, kemudian menambahkan method **tanbah()**,**tampil(),** dan **bubbleSort()**

```
public class MahasiswaBerprestasi18 {

    Mahasiswa18[] listMhs = new Mahasiswa18[5];
    int idx;

    void tambah(Mahasiswa18 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa18 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) {
                    Mahasiswa18 tmp = listMhs[j];
                    listMhs[j] = listMhs[j - 1];
                    listMhs[j - 1] = tmp;
                }
            }
        }
    }
```

3. Membuat class **MahasiswaDemo**<No Presensi> , kemudian membuat 5 objek mahasiswa dan memanggil fungsi tampil untuk menampilkan semua data, dan diurutkan emnggunakan bubbleSort()

```
  public static void main(String[] args) {
        MahasiswaBerprestasi18 list = new
MahasiswaBerprestasi18();
        Mahasiswa18 m1 = new Mahasiswa18("123",
"Zidan", "2A", 3.2);
        Mahasiswa18 m2 = new Mahasiswa18("124", "Ayu",
"2A", 3.5);
        Mahasiswa18 m3 = new Mahasiswa18("125", "Sofi",
"2A", 3.1);
        Mahasiswa18 m4 = new Mahasiswa18("126", "Sita",
"2A", 3.9);
        Mahasiswa18 m5 = new Mahasiswa18("127", "Miki",
"2A", 3.7);

        list.tambah(m1);
        list.tambah(m2);
```

2.2.2 Verifikasi Hasil Percobaan

```
Masukkan jumlah elemen: 3
Masukkan nilai basis elemen ke-1: 2
Masukkan nilai pangkat elemen ke-1: 3
Masukkan nilai basis elemen ke-2: 4
Masukkan nilai pangkat elemen ke-2: 5
Masukkan nilai basis elemen ke-3: 6
Masukkan nilai pangkat elemen ke-3: 7
HASIL PANGKAT BRUTERFORCE
2^3: 8
4^5: 1024
6^7: 279936
HASIL PANGKAT DIVIDE AND CONQUER
2^3: 8
4^5: 16384
6^7: 279936
```

2.2.3 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<listMhs.length-1 ?
      Jawab : syarat perulangan seperti kondisi tersebut agar pada saat semua kondisi sudah berurutan dengan benar maka iterasi yang paling terakhir tidak perlu dilakukan, karena semua sudah urut
   b. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?
      Jawab : syarat tersebut digunakan agar perulangan tersebut tidak Kembali mengecek elemen yang sudah urut
   c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
      Jawab : jika ada 50 data, maka perulangan akan berulang sebanyak 49 kali karena iterasi yang paling terakhir tidak perlu dilakukan, sedangkan untuk tahap bublle sort berlangsung sebanyak 49 tahap
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyborad) yang terdiri dari nim, nama, kelas, dan ipk!
   Jawab :

Kode Program

```
// Modifikasi Pertanyaan
    void input() {
        for (int i = 0; i < listMhs.length; i++) {
            System.out.println("Masukkan data Mahasiswa
ke-" + (i + 1));
            listMhs[i] = new Mahasiswa18();
            System.out.print("NIM\t: ");
            listMhs[i].nim = scan.next();
            System.out.print("Nama\t: ");
            listMhs[i].nama = scan.next();
            System.out.print("Kelas\t: ");
            listMhs[i].kelas = scan.next();
            System.out.print("IPK\t: ");
            listMhs[i].ipk = scan.nextDouble();


        }
    }
```

Hasil running kode program

```
Masukkan data Mahasiswa ke-1
NIM     : 244107020094
Nama    : Aditiasmara
Kelas   : 1E
IPK     : 999
Masukkan data Mahasiswa ke-2
NIM     : 244107020099
Nama    : Adid
Kelas   : 1F
IPK     : 899999
Masukkan data Mahasiswa ke-3
NIM     : 789921
Nama    : DImas
Kelas   : 2I
IPK     : 321
Masukkan data Mahasiswa ke-4
NIM     : 2445661
Nama    : Dafa
Kelas   : 1I
IPK     : 899291
Masukkan data Mahasiswa ke-5
NIM     : 244516161
Nama    : Ahmad
Kelas   : 1K
IPK     : 872
Data Mahasiswa setelah sorting berdasarkan IPK (DESC) :
Nama    : Adid
NIM     : 244107020099
Kelas   : 1F
IPK     : 899999.0
------------------------
Nama    : Dafa
NIM     : 2445661
Kelas   : 1I
```

2.3 Percobaan 3 : Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

2.3.1 Langkah-langkah percobaan

1. Menambahkan method selectionSort() pada kelas MahasiswaBerprestasi

```
 void SelectionSort() {
        for (int i = 0; i < listMhs.length; i++) {
            int idxMin = i;
            for (int j = i + 1; j < listMhs.length;
j++) {
                if (listMhs[j].ipk <
listMhs[idxMin].ipk) {
                    idxMin = j;
                }
            }
            Mahasiswa18 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
```

2. Menambahkan baris program pada kelas MahasiswaDemo untuk memanggil method selectionSort()

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

2.3.2 Verifikasi hasil percobaan

```
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
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
------------------------
```

2.3.3 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!
Jawab : proses tersebut berfungsi untuk mencari IPK terkecil dan mengurutkan dari IPK yang kecil ke IPK yang besar

2.4 Percobaan 4 : Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

2.4.1 Langkah-langkah percobaan

1. Menambahkan method insertionSort() pada kelas MahasiswaBerprestasi

```
void InstertionSort() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa18 temp = listMhs[i];
            int j = i;
            while (j > 0 && listMhs[j - 1].ipk >
temp.ipk) {
                listMhs[j] = listMhs[j - 1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
```

3. Menambahkan baris program pada kelas MahasiswaDemo untuk memanggil method insertionSort() dan tampil

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

2.4.2 Verifikasi hasil percobaan

```
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
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
------------------------
```

2.4.3 Pertanyaan

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

Kode program :

```
// while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
 // modifikasi pertanyaan DESC
       while (j > 0 && listMhs[j - 1].ipk < temp.ipk)
{
```

Menngubah tanda yang awalnya > menjadi <yang akan menghasilkan seperti berikut

```
Data yang sudah terurut menggunakan INSERTION SORT (DESC)
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
------------------------
```

2.5 Latihan praktikum

buatlah menu dikelas main dengan pilihan menu:
Tambah data digunakan untuk menambahkan data dosen
Tampil data digunakan untuk menampilkan data seluruh dosen
Sorting ASC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari dosen termuda ke dosen tertua menggunakan bublle Sort.
Sorting DSC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari tertua ke dosen termuda dapat menggunakan algoritma selection sort atau insertion sort.

Kelas Dosen

```
public class Dosen18 {

    String kode;
    String nama;
    boolean jenisKelamin;
    int umur;

    Dosen18() {

    }

    Dosen18(String kd, String nm, boolean jk, int age) {
        kode = kd;
        nama = nm;
        jenisKelamin = jk;
        umur = age;

    }

    void tampil(){
        System.out.println("Kode\t\t : " + kode);
        System.out.println("Nama\t\t : " + nama);
        System.out.println("Jenis Kelamin\t : " +  (jenisKelamin?
"Laki-laki" : "Perempuan"));
        System.out.println("Umur\t\t : " + umur);
    }



}
```

Kelas Dosen Data

```
import java.util.Scanner;

public class DosenData18 {

    Scanner scan = new Scanner(System.in);
    Dosen18[] listDosen = new Dosen18[10];
    int idx;

    void tambah(Dosen18 dsn) {


        if (idx >= listDosen.length) {
            System.out.println("Data telah penuh, tidak dapat
menambahkan lebih banyak dosen.");
            return;
        }
        listDosen[idx++] = dsn; // Tambahkan data dan naikkan indeks
        System.out.println("Dosen berhasil ditambahkan!");
    }

    void input (){
        for (int i = 0; i < listDosen.length; i++) {
            System.out.println("Masukkan data Dosen ke-" + (i + 1));
            listDosen[i] = new Dosen18();
            System.out.print("Kode\t\t\t: ");
            listDosen[i].kode = scan.next();
            System.out.print("Nama\t\t\t: ");
            listDosen[i].nama = scan.next();
            scan.nextLine();
            System.out.print("Jenis Kelamin(L/P)\t: ");
            String jk = scan.next();
            listDosen[i].jenisKelamin = jk.equalsIgnoreCase("L");
            System.out.print("Umur\t\t\t: ");
            listDosen[i].umur = scan.nextInt();

        }
    }
    void tampil() {
        for (Dosen18 dsn : listDosen) {
            dsn.tampil();
            System.out.println("-------------------------");
        }
    }

    void SortingASC() {
        for (int i = 0; i < listDosen.length - 1; i++) {
            for (int j = 1; j < listDosen.length - i; j++) {
                if (listDosen[j].umur < listDosen[j - 1].umur) {
                    Dosen18 temp = listDosen[j];
                    listDosen[j] = listDosen[j - 1];
                    listDosen[j - 1] = temp;

                }

            }
```

```
        }
    }

    void SortingDSC() {
        for (int i = 0; i < listDosen.length - 1; i++) {
            for (int j = 1; j < listDosen.length - i; j++) {
                if (listDosen[j].umur > listDosen[j - 1].umur) {
                    Dosen18 temp = listDosen[j];
                    listDosen[j] = listDosen[j - 1];
                    listDosen[j - 1] = temp;

                }

            }

        }
    }
}
```

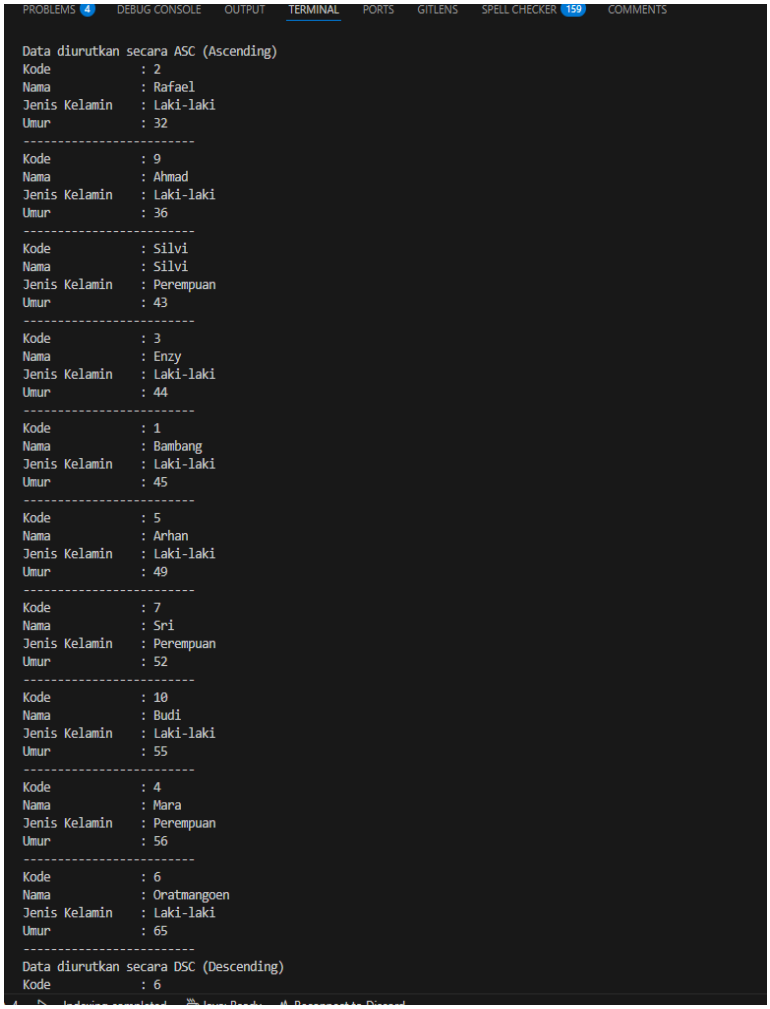
Kelas DosenMain

```
public class DosenMain18 {

    public static void main(String[] args) {
        DosenData18 list = new DosenData18();
        // jika tidak menggunakan input
        // Dosen18 d1 = new Dosen18("111", "Budi", true, 45);
        // Dosen18 d2 = new Dosen18("222", "Ari", true, 57);
        // Dosen18 d3 = new Dosen18("333", "Mawlina", false, 33);

        // list.tambah(d1);
        // list.tambah(d2);
        // list.tambah(d3);


      list.input();
        System.out.println();
        System.out.println("Tampilan Data");
        System.out.println("-----------------------------------
");

        System.out.println("Data Asli");
        list.tampil();

        System.out.println("Data diurutkan secara ASC
(Ascending)");
        list.SortingASC();
        list.tampil();

        System.out.println("Data diurutkan secara DSC
(Descending)");
        list.SortingDSC();
        list.tampil();
    }
```

Hasil output

```
Masukkan data Dosen ke-1
Kode                    : 1
Nama                    : Bambang Pamungkas
Jenis Kelamin(L/P)      : L
Umur                    : 45
Masukkan data Dosen ke-2
Kode                    : 2
Nama                    : Rafael Struick
Jenis Kelamin(L/P)      : L
Umur                    : 32
Masukkan data Dosen ke-3
Kode                    : 3
Nama                    : Enzy
Jenis Kelamin(L/P)      : L
Umur                    : 44
Masukkan data Dosen ke-4
Kode                    : 4
Nama                    : Mara Salva
Jenis Kelamin(L/P)      : P
Umur                    : 56
Masukkan data Dosen ke-5
Kode                    : 5
Nama                    : Arhan Pratama
Jenis Kelamin(L/P)      : L
Umur                    : 49
Masukkan data Dosen ke-6
Kode                    : 6
Nama                    : Oratmangoen
Jenis Kelamin(L/P)      : L
Umur                    : 65
Masukkan data Dosen ke-7
Kode                    : 7
Nama                    : Sri
Jenis Kelamin(L/P)      : P
Umur                    : 52
Masukkan data Dosen ke-8
Kode                    : Silvi
Nama                    : Silvi
Jenis Kelamin(L/P)      : P
Umur                    : 43
Masukkan data Dosen ke-9
Kode                    : 9
Nama                    : Ahmad
Jenis Kelamin(L/P)      : L
Umur                    : 36
Masukkan data Dosen ke-10
Kode                    : 10
Nama                    : Budi
Jenis Kelamin(L/P)      : L
Umur                    : 55
```

```
Tampilan Data
-------------------------------------
Data Asli
Kode            : 1
Nama            : Bambang
Jenis Kelamin   : Laki-laki
Umur            : 45
-------------------------
Kode            : 2
Nama            : Rafael
Jenis Kelamin   : Laki-laki
Umur            : 32
-------------------------
Kode            : 3
Nama            : Enzy
Jenis Kelamin   : Laki-laki
Umur            : 44
-------------------------
Kode            : 4
Nama            : Mara
Jenis Kelamin   : Perempuan
Umur            : 56
-------------------------
Kode            : 5
Nama            : Arhan
Jenis Kelamin   : Laki-laki
Umur            : 49
-------------------------
Kode            : 6
Nama            : Oratmangoen
Jenis Kelamin   : Laki-laki
Umur            : 65
-------------------------
Kode            : 7
Nama            : Sri
Jenis Kelamin   : Perempuan
Umur            : 52
-------------------------
Kode            : Silvi
Nama            : Silvi
Jenis Kelamin   : Perempuan
Umur            : 43
-------------------------
```


PROBLEMS 4
DEBUG CONSOLE
OUTPUT
TERMINAL
PORTS
GITLENS
SPELL CHECKER 159
COMMENTS
Data diurutkan secara ASC (Ascending)
Kode : 2
Nama : Rafael
Jenis Kelamin : Laki-laki
Umur : 32
Kode : 9
Nama : Ahmad
Jenis Kelamin : Laki-laki
Umur : 36
Kode : Silvi
Nama : Silvi
Jenis Kelamin : Perempuan
Umur : 43
Kode : 3
Nama : Enzy
Jenis Kelamin : Laki-laki
Umur : 44
Kode : 1
Nama : Bambang
Jenis Kelamin : Laki-laki
Umur : 45
Kode : 5
Nama : Arhan
Jenis Kelamin : Laki-laki
Umur : 49
Kode : 7
Nama : Sri
Jenis Kelamin : Perempuan
Umur : 52
Kode : 10
Nama : Budi
Jenis Kelamin : Laki-laki
Umur : 55
Kode : 4
Nama : Mara
Jenis Kelamin : Perempuan
Umur : 56
Kode : 6
Nama : Oratmangoen
Jenis Kelamin : Laki-laki
Umur : 65
Data diurutkan secara DSC (Descending)
Kode : 6

```
------------------------
Data diurutkan secara DSC (Descending)
Kode            : 6
Nama            : Oratmangoen
Jenis Kelamin   : Laki-laki
Umur            : 65
------------------------
Kode            : 4
Nama            : Mara
Jenis Kelamin   : Perempuan
Umur            : 56
------------------------
Kode            : 10
Nama            : Budi
Jenis Kelamin   : Laki-laki
Umur            : 55
------------------------
Kode            : 7
Nama            : Sri
Jenis Kelamin   : Perempuan
Umur            : 52
------------------------
Kode            : 5
Nama            : Arhan
Jenis Kelamin   : Laki-laki
Umur            : 49
------------------------
Kode            : 1
Nama            : Bambang
Jenis Kelamin   : Laki-laki
Umur            : 45
------------------------
Kode            : 3
Nama            : Enzy
Jenis Kelamin   : Laki-laki
Umur            : 44
------------------------
Kode            : Silvi
Nama            : Silvi
Jenis Kelamin   : Perempuan
Umur            : 43
------------------------
Kode            : 9
Nama            : Ahmad
Jenis Kelamin   : Laki-laki
Umur            : 36
------------------------
Kode            : 2
Nama            : Rafael
Jenis Kelamin   : Laki-laki
Umur            : 32
Umur            : 43
```