

Laporan Hasil Praktikum Algoritma Dan Struktur Data
Jobsheet 6



Disusun Oleh :

Nama : Fadhil Taufiqurrachman
NIM : 244107020090
Kelas : Teknik Informatika 1E

Program Studi Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang 2025

6.2 Praktikum 1 – Mengimplementasikan Sorting Menggunakan Object

6.2.1 Percobaan 1 : Sorting – Bubble Sort

6.2.1.1 Kode Program

Kode program pada class Sorting08 :

```
package Jobsheet6;

public class Sorting08 {
    int [] data;
    int jumData;

    Sorting08(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();
    }
}
```

Kode program pada class SortingMain08 :

```
package Jobsheet6;

public class SortingMain08 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        Sorting08 dataurut1 = new Sorting08(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();
    }
}
```

6.2.1.2 Verifikasi

```
Data Awal 1
20 10 2 7 12
Data Sudah Diurutkan Dengan Bubble Sort (ASC)
2 7 10 12 20
```

6.2.2 Percobaan 2 : Sorting – Selection Sort

6.2.2.1 Kode Program

Menambahkan kode program berikut pada class Sorting08 :

```
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;
    }
}
```

Menambahkan kode program pada class SortingMain08 :

```
int b[] = {30, 20, 2, 8, 14};
Sorting08 dataurut2 = new Sorting08(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();
```

6.2.2.2 Verifikasi

```
Data Awal 2
30 20 2 8 14
Data Sudah Diurutkan Dengan Selection Sort (ASC)
2 8 14 20 30
```

6.2.3 Percobaan 3 : Sorting – Insertion Sort

6.2.3.1 Kode Program

Menambahkan kode program berikut pada class Sorting08 :

```
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;
    }
}
```

Menambahkan kode program pada class SortingMain08 :

```
int c[] = {40, 10, 4, 9, 3};
Sorting08 dataurut3 = new Sorting08(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();
```

6.2.3.2 Verifikasi

```
Data Awal 3
40 10 4 9 3
Data Sudah Diurutkan Dengan Insertion Sort (ASC)
3 4 9 10 40
```

6.2.4 Pertanyaan

1. Kode program di bawah ini terdapat pada method `bubbleSort()`.

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

Fungsi dari kode program tersebut adalah untuk perbandingan dengan data berikutnya, yang menurut metode Bubble Sort yaitu membandingkan data pada indeks 0 dengan data pada indeks 1, kemudian data pada indeks 1 dibandingkan dengan data pada indeks 2 dan seterusnya sampai sejumlah datanya. Jika kondisinya benar dan terpenuhi, maka akan melakukan proses swap dengan cara mengcopy nilai data yang akan dipindahkan ke variabel baru untuk sementara. Kemudian mengganti nilai data dari indeks yang dituju dengan nilai yang ingin di swap.

2. Berikut adalah kode program yang merupakan algoritma pencarian nilai minimum pada selection sort.

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

3. Loop ini digunakan untuk membandingkan nilai pada variabel temp dengan elemen-elemen sebelumnya, dalam hal ini variabel j diinisialisasi dengan i - 1, yang merupakan indeks elemen sebelumnya. Loop ini akan terus berjalan jika kondisi tersebut terpenuhi, yaitu selama masih ada elemen sebelumnya untuk dibandingkan dan elemen sebelumnya lebih besar dari temp.
4. Jika kondisi loop while terpenuhi, maka elemen data[j] akan dipindahkan atau dicopy satu posisi ke kanan yakni ke data[j + 1]. Hal ini akan membuat ruang untuk menyisipkan temp pada posisi yang seharusnya.

6.2.5 Commit Dan Push Ke Akun GitHub

```
PS C:\File Kuliah\Semester 2\Praktikum Algoritma Dan Struktur Data\GitHub\Praktikum-ASD\Jobsheet6> git add Sorting08.java SortingMain08.java
PS C:\File Kuliah\Semester 2\Praktikum Algoritma Dan Struktur Data\GitHub\Praktikum-ASD\Jobsheet6> git commit -m "Praktikum 1"
[main cb6b70b] Praktikum 1
 2 files changed, 91 insertions(+)
 create mode 100644 Jobsheet6/Sorting08.java
 create mode 100644 Jobsheet6/SortingMain08.java
PS C:\File Kuliah\Semester 2\Praktikum Algoritma Dan Struktur Data\GitHub\Praktikum-ASD\Jobsheet6> git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.07 KiB | 22.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/FadhilTaufiqurrachman/Praktikum-ASD.git
 63b1ad4..cb6b70b  main -> main
```

6.3 Praktikum 2 – Mengimplementasikan Sorting Menggunakan Array Of Object

6.3.1 Percobaan 1 : Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

6.3.1.1 Kode Program

Kode program pada class Mahasiswa08 :

```
package Jobsheet6;

public class Mahasiswa08 {
    String nim, nama, kelas;
    double ipk;

    public Mahasiswa08() {

    }

    public Mahasiswa08(String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }

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

Kode program pada class MahasiswaBerprestasi08 :

```
package Jobsheet6;

public class MahasiswaBerprestasi08 {
    Mahasiswa08 listMhs[] = new Mahasiswa08[5];
    int idx;

    void tambah(Mahasiswa08 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data Sudah Penuh!");
        }
    }

    void tampil() {
        for (Mahasiswa08 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) {
                    Mahasiswa08 tmp = listMhs[j];
                    listMhs[j] = listMhs[j - 1];
                    listMhs[j - 1] = tmp;
                }
            }
        }
    }
}
```


Kode program pada class MahasiswaDemo08 :

```
package Jobsheet6;

public class MahasiswaDemo08 {
    public static void main(String[] args) {
        MahasiswaBerprestasi08 list = new MahasiswaBerprestasi08();
        Mahasiswa08 m1 = new Mahasiswa08("123", "Zidan", "2A", 3.2);
        Mahasiswa08 m2 = new Mahasiswa08("124", "Ayu", "2A", 3.5);
        Mahasiswa08 m3 = new Mahasiswa08("125", "Sofi", "2A", 3.1);
        Mahasiswa08 m4 = new Mahasiswa08("126", "Sita", "2A", 3.9);
        Mahasiswa08 m5 = new Mahasiswa08("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();
    }
}
```

6.3.1.2 Verifikasi

Data Mahasiswa Sebelum Sorting :

NIM : 123
Nama : Zidan
Kelas : 2A
IPK : 3.2

NIM : 124
Nama : Ayu
Kelas : 2A
IPK : 3.5

NIM : 125
Nama : Sofi
Kelas : 2A
IPK : 3.1

NIM : 126
Nama : Sita
Kelas : 2A
IPK : 3.9

NIM : 127
Nama : Miki
Kelas : 2A
IPK : 3.7

Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC) :

NIM : 126
Nama : Sita
Kelas : 2A
IPK : 3.9

NIM : 127
Nama : Miki
Kelas : 2A
IPK : 3.7

NIM : 124
Nama : Ayu
Kelas : 2A
IPK : 3.5

NIM : 123
Nama : Zidan
Kelas : 2A
IPK : 3.2

NIM : 125
Nama : Sofi
Kelas : 2A
IPK : 3.1

6.3.1.3 Pertanyaan

1. Berikut ini adalah kode perulangan di dalam method `bubbleSort()`.

```
for (int i = 0; i < listMhs.length - 1; i++) {  
    for (int j = 1; j < listMhs.length - i; j++) {
```

- a) Karena jika sudah melakukan iterasi sejumlah $\text{length} - 1$, hanya tersisa 1 nilai elemen di paling kiri. Maka dari itu tidak perlu membandingkannya lagi karena sudah tidak ada lagi bilangan pembandingnya dan itu sudah pasti berada pada posisi yang tepat.
- b) Karena setiap kali perulangan i selesai, nilai pada array indeks terakhir sudah tidak perlu di cek karena sudah pasti benar, atau bisa dibilang nilai akhir dikunci tidak perlu diikuti lagi. Jika sudah, maka pada iterasi selanjutnya yang dikunci adalah 2 nilai terakhir dari kanan array, lalu bertambah lagi menjadi 3 pada iterasi berikutnya dan seterusnya sampai perulangan i selesai.
- c) Akan berlangsung sebanyak 49 kali.

2. Berikut adalah penambahan kode program pada class Mahasiswa08.

```
import java.util.Scanner;
void tambahData() {
    Scanner sc = new Scanner(System.in);
    System.out.print("NIM      : ");
    nim = sc.nextLine();
    System.out.print("Nama      : ");
    nama = sc.nextLine();
    System.out.print("Kelas   : ");
    kelas = sc.nextLine();
    System.out.print("IPK      : ");
    ipk = sc.nextDouble();
    sc.nextLine();
}
```

Kemudian memodifikasi input pada class MahasiswaDemo08 yang tadinya input statis menjadi input dinamis. Berikut adalah modifikasi kode programnya.

```
MahasiswaBerprestasi08 list = new MahasiswaBerprestasi08();
for (int i = 0; i < 5; i++) {
    System.out.println("Masukkan Data Mahasiswa Ke-" + (i + 1));
    Mahasiswa08 m = new Mahasiswa08();
    m.tambahData();
    list.tambah(m);
    System.out.println("-----");
}
```

6.3.2 Percobaan 2 : Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

6.3.2.1 Kode Program

Menambahkan kode program berikut pada class MahasiswaBerprestasi08 :

```
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;
            }
        }
        Mahasiswa08 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

Menambahkan kode program pada class MahasiswaDemo08 :

```
System.out.println("Data Yang Sudah Terurut Menggunakan  
Selection Sort (ASC) : ");  
list.selectionSort();  
list.tampil();
```

6.3.2.2 Verifikasi

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

6.3.2.3 Pertanyaan

1. Berikut ini adalah kode di dalam method `selectionSort()`.

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

Variabel `idxMin` diinisialisasi sama dengan `i`. Ini mengasumsikan atau mengatur bahwa elemen pada indeks ke `i` adalah elemen minimum dalam bagian array yang belum diurutkan. Yang kemudian dilanjutkan dengan perulangan yang merupakan search untuk mencari apakah ada elemen disebelah kanan indeks ke `idxMin` yang sesuai dengan kondisi `if` di dalam perulangan tersebut. Jika kondisi terpenuhi, maka nilai `idxMin` akan diganti dan diinisialisasikan sama dengan `j`.

6.3.3 Percobaan 3 : Mengurutkan Data Mahasiswa Berdasarkan IPK (Insertion Sort)

6.3.3.1 Kode Program

Menambahkan kode program berikut pada class MahasiswaBerprestasi08 :

```
void insertionSort() {  
    for (int i = 1; i < listMhs.length; i++) {  
        Mahasiswa08 temp = listMhs[i];  
        int j = i;  
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {  
            listMhs[j] = listMhs[j - 1];  
            j--;  
        }  
        listMhs[j] = temp;  
    }  
}
```

Menambahkan kode program pada class MahasiswaDemo08 :

```
        System.out.println("Data Yang Sudah Terurut Menggunakan  
Insertion Sort (ASC) : ");  
        list.insertionSort();  
        list.tampil();
```

6.3.3.2 Verifikasi

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

6.3.3.3 Pertanyaan

1. Modifikasi kode program di dalam method `insertionSort()`. Yaitu kita hanya perlu mengubah kondisi perulangan yang awalnya lebih dari menjadi kurang dari.

```
void insertionSort() {  
    for (int i = 1; i < listMhs.length; i++) {  
        Mahasiswa08 temp = listMhs[i];  
        int j = i;  
        while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {  
            listMhs[j] = listMhs[j - 1];  
            j--;  
        }  
        listMhs[j] = temp;  
    }  
}
```

Berikut adalah hasilnya.

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 : Yayuk

Kelas : 2C

IPK : 3.4

NIM : 444

Nama : Susi

Kelas : 2C

IPK : 3.1

NIM : 222

Nama : Dika

Kelas : 2C

IPK : 3.0

6.4 Latihan Praktikum

6.4.1 Kode Program

Kode program pada class Dosen08 :

```
package Jobsheet6;

public class Dosen08 {
    String kode, nama;
    boolean jenisKelamin;
    int usia;

    Dosen08(String kd, String name, boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

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

Kode program pada class DataDosen08 :

```
package Jobsheet6;

public class DataDosen08 {
    Dosen08 dataDosen[] = new Dosen08[10];
    int idx;

    void tambah(Dosen08 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data Sudah Penuh!");
        }
    }

    void tampil() {
        if (dataDosen[0] == null) {
            System.out.println("Data Kosong!");
        } else {
            System.out.println("Data Dosen : ");
            for (Dosen08 dsn : dataDosen) {
                dsn.tampilkanData();
            }
        }
    }

    void sortingASC() {
        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) {
                    Dosen08 dummy = dataDosen[j];
                    dataDosen[j] = dataDosen[j - 1];
                    dataDosen[j - 1] = dummy;
                }
            }
        }
    }

    void sortingDSC() {
        for (int i = 0; i < dataDosen.length; i++) {
            Dosen08 temp = dataDosen[i];
            int j = i;
            while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j - 1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}
```

Kode program pada class DataDosen08 :

```
package Jobsheet6;

import java.util.Scanner;
public class DosenMain08 {
    Scanner input = new Scanner(System.in);
    static int menu;
    static DataDosen08 data = new DataDosen08();

    void menampilkanMenu() {
        System.out.println("=====");
        System.out.println("===== Pilihan Menu =====");
        System.out.println("1. Tambah Data Dosen.\n" +
            "2. Tampil Data Dosen.          \n" +
            "3. Sorting ASC Usia Dosen.         \n" +
            "4. Sorting DSC Usia Dosen.         \n" +
            "5. Keluar.                          ");
        System.out.println("=====");
        System.out.print("Masukkan Nomor Menu : ");
        menu = input.nextInt();
        input.nextLine();
        System.out.println();
    }

    void menambahData() {
        for (int i = 0; i < 10; i++) {
            System.out.println("Masukkan Data Dosen Ke-" + (i + 1));
            System.out.print("Masukkan Kode Dosen      : ");
            String kode = input.nextLine();
            System.out.print("Masukkan Nama Dosen      : ");
            String nama = input.nextLine();
            System.out.print("Jenis Kelamin L (true/false)? : ");
            boolean jenisKelamin = input.nextBoolean();
            System.out.print("Masukkan Usia Dosen      : ");
            int usia = input.nextInt();
            input.nextLine();
            System.out.println("-----");
            Dosen08 dsn = new Dosen08(kode, nama, jenisKelamin, usia);
            data.tambah(dsn);
        }
    }

    void menampilkanDataDosen() {
        data.tampil();
    }
}
```

```

void mengurutkanASC() {
    data.sortingASC();
    data.tampil();
}

void mengurutkanDSC() {
    data.sortingDSC();
    data.tampil();
}

public static void main(String[] args) {
    do {
        DosenMain08 app = new DosenMain08();
        app.menampilkanMenu();
        switch (app.menu) {
            case 1:
                app.menambahData();
                break;
            case 2:
                app.menampilkanDataDosen();
                break;
            case 3:
                app.mengurutkanASC();
                break;
            case 4:
                app.mengurutkanDSC();
                break;
            case 5:
                System.out.println("Terima Kasih!");
                break;
            default:
                System.out.println("Nomor Menu Tidak Ada!");
                break;
        }
    } while (menu != 5);
}
}

```

6.4.2 Hasil

```
===== Pilihan Menu =====
```

1. Tambah Data Dosen.
2. Tampil Data Dosen.
3. Sorting ASC Usia Dosen.
4. Sorting DSC Usia Dosen.
5. Keluar.

```
=====
```

```
Masukkan Nomor Menu : 1
```

```
Masukkan Data Dosen Ke-1
```

```
Masukkan Kode Dosen      : 1
Masukkan Nama Dosen      : Afa
Jenis Kelamin L (true/false)? : true
Masukkan Usia Dosen      : 20
```

```
-----
Masukkan Data Dosen Ke-2
```

```
Masukkan Kode Dosen      : 2
Masukkan Nama Dosen      : Fira
Jenis Kelamin L (true/false)? : false
Masukkan Usia Dosen      : 25
```

```
-----
Masukkan Data Dosen Ke-3
```

```
Masukkan Kode Dosen      : 5
Masukkan Nama Dosen      : Alif
Jenis Kelamin L (true/false)? : true
Masukkan Usia Dosen      : 39
```

```
-----
Masukkan Data Dosen Ke-4
```

```
Masukkan Kode Dosen      : 12
Masukkan Nama Dosen      : Hana
Jenis Kelamin L (true/false)? : false
Masukkan Usia Dosen      : 23
```

```
-----
Masukkan Data Dosen Ke-5
```

```
Masukkan Kode Dosen      : 18
Masukkan Nama Dosen      : Jamal
Jenis Kelamin L (true/false)? : true
Masukkan Usia Dosen      : 45
```

```
-----  
Masukkan Data Dosen Ke-6  
Masukkan Kode Dosen      : 29  
Masukkan Nama Dosen      : Ayu  
Jenis Kelamin L (true/false)? : false  
Masukkan Usia Dosen      : 35  
-----  
Masukkan Data Dosen Ke-7  
Masukkan Kode Dosen      : 40  
Masukkan Nama Dosen      : Fafa  
Jenis Kelamin L (true/false)? : true  
Masukkan Usia Dosen      : 29  
-----  
Masukkan Data Dosen Ke-8  
Masukkan Kode Dosen      : 55  
Masukkan Nama Dosen      : Tina  
Jenis Kelamin L (true/false)? : false  
Masukkan Usia Dosen      : 50  
-----  
Masukkan Data Dosen Ke-9  
Masukkan Kode Dosen      : 44  
Masukkan Nama Dosen      : Toni  
Jenis Kelamin L (true/false)? : true  
Masukkan Usia Dosen      : 33  
-----  
Masukkan Data Dosen Ke-10  
Masukkan Kode Dosen      : 63  
Masukkan Nama Dosen      : Rika  
Jenis Kelamin L (true/false)? : false  
Masukkan Usia Dosen      : 48  
-----
```



```
=====
===== Pilihan Menu =====
```

1. Tambah Data Dosen.
2. Tampil Data Dosen.
3. Sorting ASC Usia Dosen.
4. Sorting DSC Usia Dosen.
5. Keluar.

```
=====
Masukkan Nomor Menu : 2
```

Data Dosen :

Kode Dosen : 1
Nama Dosen : Afwa
Jenis Kelamin : Laki-laki
Usia : 20

Kode Dosen : 2
Nama Dosen : Fira
Jenis Kelamin : Perempuan
Usia : 25

Kode Dosen : 5
Nama Dosen : Alif
Jenis Kelamin : Laki-laki
Usia : 39

Kode Dosen : 12
Nama Dosen : Hana
Jenis Kelamin : Perempuan
Usia : 23

Kode Dosen : 18
Nama Dosen : Jamal
Jenis Kelamin : Laki-laki
Usia : 45

Kode Dosen : 29

```
=====
===== Pilihan Menu =====
1. Tambah Data Dosen.
2. Tampil Data Dosen.
3. Sorting ASC Usia Dosen.
4. Sorting DSC Usia Dosen.
5. Keluar.
```

```
=====
Masukkan Nomor Menu : 3
```

Data Dosen :

```
Kode Dosen      : 1
Nama Dosen      : Afwa
Jenis Kelamin   : Laki-laki
Usia            : 20
```

```
-----
Kode Dosen      : 12
Nama Dosen      : Hana
Jenis Kelamin   : Perempuan
Usia            : 23
```

```
-----
Kode Dosen      : 2
Nama Dosen      : Fira
Jenis Kelamin   : Perempuan
Usia            : 25
```

```
-----
Kode Dosen      : 40
Nama Dosen      : Fafa
Jenis Kelamin   : Laki-laki
Usia            : 29
```

```
-----
Kode Dosen      : 44
Nama Dosen      : Toni
Jenis Kelamin   : Laki-laki
Usia            : 33
```

```
-----
Kode Dosen      : 29
Nama Dosen      : Ayu
```

```
=====
===== Pilihan Menu =====
```

1. Tambah Data Dosen.
2. Tampil Data Dosen.
3. Sorting ASC Usia Dosen.
4. Sorting DSC Usia Dosen.
5. Keluar.

```
=====
Masukkan Nomor Menu : 4
```

Data Dosen :

Kode Dosen : 55
Nama Dosen : Tina
Jenis Kelamin : Perempuan
Usia : 50

Kode Dosen : 63
Nama Dosen : Rika
Jenis Kelamin : Perempuan
Usia : 48

Kode Dosen : 18
Nama Dosen : Jamal
Jenis Kelamin : Laki-laki
Usia : 45

Kode Dosen : 5
Nama Dosen : Alif
Jenis Kelamin : Laki-laki
Usia : 39

Kode Dosen : 29
Nama Dosen : Ayu
Jenis Kelamin : Perempuan
Usia : 35

Kode Dosen : 44
Nama Dosen : Toni

Kode Dosen : 44
Nama Dosen : Toni
Jenis Kelamin : Laki-laki
Usia : 33

Kode Dosen : 40
Nama Dosen : Fafa
Jenis Kelamin : Laki-laki
Usia : 29

Kode Dosen : 2
Nama Dosen : Fira
Jenis Kelamin : Perempuan
Usia : 25

Kode Dosen : 12
Nama Dosen : Hana
Jenis Kelamin : Perempuan
Usia : 23

Kode Dosen : 1
Nama Dosen : Afa
Jenis Kelamin : Laki-laki
Usia : 20

=====
===== Pilihan Menu =====
1. Tambah Data Dosen.
2. Tampil Data Dosen.
3. Sorting ASC Usia Dosen.
4. Sorting DSC Usia Dosen.
5. Keluar.

=====
Masukkan Nomor Menu : 5

Terima Kasih!