

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN  
STRUKTUR DATA  
JOBSHEET 6**



**NAMA : MOHAMAT FAUZI ROHMAN**

**NIM : 244107020067**

**KELAS : 1E**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2025**

## JOBSHEET 6

### 6. Praktikum

#### 6.1 Praktikum 1 - Mengimplementasikan Sorting menggunakan object

##### 6.1.1 Langkah-langkah Percobaan

###### a. SORTING – BUBBLE SORT

1. Buat class Sorting<No Presensi>, kemudian tambahkan konstruktor dengan parameter Data[] dan jmlDat

```
public class Sorting16 {  
  
    int [] data;  
    int jumData;  
  
    Sorting16 (int Data[], int jmlDat){  
        jumData = jmlDat;  
        data = new int[jmlDat];  
        for (int i = 0; i < jumData; i++){  
            data[i] = Data[i];  
        }  
    }  
}
```

2. Buatlah method bubbleSort bertipe void dan deklarasikan isinya menggunakan algoritma Bubble Sort.

```
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;  
            }  
        }  
    }  
}
```

3. Buatlah method tampil bertipe void dan deklarasikan isi method tersebut

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

4. Buat class SortingMain kemudian deklarasikan array dengan nama a[] kemudian isi array tersebut

```
public class SortingMain16 {  
    public static void main(String[] args) {  
  
        int a[] = {20, 10, 2, 7, 12};  
    }  
}
```

5. Buatlah objek baru dengan nama `dataurut1` yang merupakan instansiasi dari class `Sorting`, kemudian isi parameternya. Dan lakukan pemanggilan method `bubbleSort` dan `tampil`

```
Sorting16 dataurut1 = new Sorting16(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();
    }
}
```

#### 6.1.2 Verifikasi Hasil Percobaan

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

#### b. SORTING – SELECTION SORT

1. Pada class `Sorting16` yang sudah dibuat di praktikum sebelumnya tambahkan method `SelectionSort` yang mengimplementasikan pengurutan menggunakan algoritma selection sort

```
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;
    }
}
```

2. Deklarasikan array dengan nama `b[]` pada kelas `SortingMain` kemudian isi array tersebut

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

3. Buatlah objek baru dengan nama `dataurut2` yang merupakan instansiasi dari class `Sorting16`, kemudian isi parameternya dan lakukan pemanggilan method `SelectionSort` dan `tampil`

```
Sorting16 dataurut2 = new Sorting16(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();
System.out.println();
```

#### 6.1.3 Verifikasi Hasil Percobaan

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

#### c. SORTING – INSERTION SORT

1. Pada class Sorting16 yang sudah dibuat di praktikum sebelumnya tambahkan method insertionSort yang mengimplementasikan pengurutan menggunakan algoritma insertion sort

```
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. Deklarasikan array dengan nama c[] pada kelas SortingMain kemudian isi array tersebut

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

3. Buatlah objek baru dengan nama dataurut3 yang merupakan instansiasi dari class Sorting16, kemudian isi parameternya dan lakukan pemanggilan method insertionSort dan tampil

```
Sorting16 dataurut3 = new Sorting16(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();
}
```

#### 6.1.4 Verifikasi Hasil Percobaan

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

#### 6.1.5 Pertanyaan

1. Kode tersebut adalah proses Bubble Sort dengan pengurutan secara Ascending. Jika data[j-1] lebih besar dibandingkan data[j], maka data[j] akan disimpan sementara pada variabel temp dan data[j] akan menempati posisi data[j-1] begitupun data[j-1] akan menempati posisi data[j]

2.

```
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;
    }
}
```

3. Untuk mencari posisi yang tepat untuk menyisipkan nilai temp ke dalam array. Dengan memindahkan elemen yang lebih besar ke kanan, agar temp bisa disisipkan di tempat yang benar.
4. Digunakan untuk memindahkan elemen yang lebih besar ke kanan dan memberikan ruang untuk menyisipkan temp ke dalam posisi yang benar dalam array

## 6.2 Praktikum 2- (Sorting Menggunakan Array of Object)

### 6.2.1 Langkah-langkah Percobaan

1. Buatlah class dengan nama Mahasiswa dan untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini

```
public class Mahasiswa16 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa16(){

    }

    Mahasiswa16(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);
    }
}
```

2. Buat class MahasiswaBerprestasi dan Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.

```
public class MahasiswaBerprestasi16 {
    Mahasiswa16 [] listMhs = new Mahasiswa16[5];
    int idx;

    void tambah (Mahasiswa16 m){
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        }else {
            System.out.println("Data sudah penuh");
        }
    }
}
```

3. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.

```
void tampil(){
    for (Mahasiswa16 m : listMhs){
        m.tampilInformasi();
        System.out.println("-----");
    }
}
```

4. Tambahkan method bubbleSort() di dalam class tersebut!

```
void bubbleSort() {
    int temp = 0;
    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) {
                Mahasiswa16 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}
```

5. Buat class MahasiswaDemo, kemudian buatlah sebuah objek MahasiswaBerprestasi dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek MahasiswaBerprestasi. Silakan dipanggil fungsi tampil() untuk melihat semua data yang telah dimasukan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.

```
public class MahasiswaDemo16 {
    public static void main(String[] args) {
        MahasiswaBerprestasi16 list = new MahasiswaBerprestasi16();
        Mahasiswa16 m1 = new Mahasiswa16("123", "Zidan", "2A", 3.2);
        Mahasiswa16 m2 = new Mahasiswa16("124", "Ayu", "2A", 3.5);
        Mahasiswa16 m3 = new Mahasiswa16("125", "Sofi", "2A", 3.1);
        Mahasiswa16 m4 = new Mahasiswa16("126", "Sita", "2A", 3.9);
        Mahasiswa16 m5 = new Mahasiswa16("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();
    }
}
```

### 6.2.2 Verifikasi Hasil Percobaan

<pre>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 -----</pre>	<pre>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 -----</pre>
--	--

### 6.2.3 Pertanyaan

1. a. Karena dalam bubble sort digunakan untuk melakukan perbandingan sebanyak jumlah elemen-1. Setelah semua elemen dilakukan, maka elemen terakhir tidak perlu dibandingkan karena sudah pasti berada di tempat yang benar.  
  
b. Digunakan untuk mengurangi panjang listMhs dengan i, karena setiap menyelesaikan satu iterasi, maka elemen terbesar sudah berada di posisi akhir. Jadi, tidak perlu memeriksa elemen di posisi akhir  
  
c. Banyak perulangan i dan tahap yang dilakukan bubble sort ialah 49 kali

### 2. Mahasiswa16

```
public class Mahasiswa16 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa16() {

    }

    Mahasiswa16(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);
    }
}
```



### MahasiswaBerprestasi16

```
public class MahasiswaBerprestasi16 {
    Mahasiswa16 [] listMhs = new Mahasiswa16[5];
    int idx;

    void tambah (Mahasiswa16 m){
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        }else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil(){
        for (Mahasiswa16 m : listMhs){
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    void bubbleSort(){
        int temp = 0;
        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) {
                    Mahasiswa16 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}
```

### MahasiswaDemo16

```
import java.util.Scanner;
public class MahasiswaDemo16 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        MahasiswaBerprestasi16 list = new MahasiswaBerprestasi16();

        for (int i = 0; i < 5; i++) {
            System.out.println("Mahasiswa ke-" + (i + 1) + ":");
            System.out.print("NIM: ");
            String nim = sc.nextLine();
            System.out.print("Nama: ");
            String nama = sc.nextLine();
            System.out.print("Kelas: ");
            String kelas = sc.nextLine();
            System.out.print("IPK: ");
            double ipk = sc.nextDouble();
            sc.nextLine();
        }
    }
}
```

```

Mahasiswa16 mhs = new Mahasiswa16(nim, nama, kelas, ipk);
    list.tambah(mhs);
}
System.out.println();
System.out.println("Data mahasiswa sebelum sorting: ");
list.tampil();

System.out.println();
System.out.println("Data Mahasiswa setelah Sorting berdasarkan IPK (DESC):
");
list.bubbleSort();
list.tampil();
}
}

```

## Hasil

<p>Mahasiswa ke-1:  NIM: 2441  Nama: Fauzi  Kelas: 2E  IPK: 3.5</p> <p>Mahasiswa ke-2:  NIM: 2442  Nama: Agus  Kelas: 2A  IPK: 3.6</p> <p>Mahasiswa ke-3:  NIM: 2443  Nama: Hadi  Kelas: 2I  IPK: 3.3</p> <p>Mahasiswa ke-4:  NIM: 2444  Nama: Gita  Kelas: 2C  IPK: 3.1</p> <p>Mahasiswa ke-5:  NIM: 2445  Nama: Yuli  Kelas: 2A  IPK: 3.6</p>	<p>Data mahasiswa sebelum sorting:</p> <p>Nama : Fauzi  NIM : 2441  Kelas: 2E  IPK : 3.5</p> <p>-----</p> <p>Nama : Agus  NIM : 2442  Kelas: 2A  IPK : 3.6</p> <p>-----</p> <p>Nama : Hadi  NIM : 2443  Kelas: 2I  IPK : 3.3</p> <p>-----</p> <p>Nama : Gita  NIM : 2444  Kelas: 2C  IPK : 3.1</p> <p>-----</p> <p>Nama : Yuli  NIM : 2445  Kelas: 2A  IPK : 3.6</p> <p>-----</p>
---	---

```

Data Mahasiswa setelah Sorting berdasarkan IPK (DESC):
Nama : Agus
NIM : 2442
Kelas: 2A
IPK : 3.6
-----
Nama : Yuli
NIM : 2445
Kelas: 2A
IPK : 3.6
-----
Nama : Fauzi
NIM : 2441
Kelas: 2E
IPK : 3.5
-----
Nama : Hadi
NIM : 2443
Kelas: 2I
IPK : 3.3
-----
Nama : Gita
NIM : 2444
Kelas: 2C
IPK : 3.1
-----

```

## 6.3 Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

### 6.3.1 Langkah-langkah Percobaan

1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya! Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan selection sort.

```

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;
            }
        }
        Mahasiswa16 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}

```

2. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method selectionSort() tersebut, kemudian panggil method tampil() untuk menampilkan data yang sudah diurutkan!

```

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

```

### 6.3.2 Verifikasi Hasil Percobaan

```
Mahasiswa ke-1:
NIM: 123
Nama: Ali
Kelas: 2B
IPK: 3.9
-----
Mahasiswa ke-2:
NIM: 124
Nama: Ila
Kelas: 2B
IPK: 3.1
-----
Mahasiswa ke-3:
NIM: 125
Nama: Agus
Kelas: 2B
IPK: 3.6
-----
Mahasiswa ke-4:
NIM: 126
Nama: Tika
Kelas: 2B
IPK: 3.3
-----
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
-----
```

### 6.3.3 Pertanyaan

1. Digunakan untuk menemukan indeks dari elemen dengan nilai terkecil, yang dimana idxMin digunakan untuk menyimpan indeks sementara dari elemen yang terkecil

## 6.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

### 6.4.1 Langkah-langkah Percobaan

1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method insertionSort() di dalamnya. Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan Insertion Sort

```
void insertionSort(){
    for (int i = 0; i < listMhs.length; i++){
        Mahasiswa16 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. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() dan tampil () tersebut!

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

3. Coba jalankan kembali class MahasiswaDemo, dan amati hasilnya! Apakah kini data mahasiswa telah tampilurut menaik berdasar ipk?

#### 6.4.2 Verifikasi Hasil Percobaan

```
Mahasiswa ke-1
NIM: 111
Nama: ayu
Kelas: 2c
IPK: 3.7
-----
Mahasiswa ke-2
NIM: 222
Nama: dika
Kelas: 2c
IPK: 3.0
-----
Mahasiswa ke-3
NIM: 333
Nama: ila
Kelas: 2c
IPK: 3.8
-----
Mahasiswa ke-4
NIM: 444
Nama: susi
Kelas: 2c
IPK: 3.1
-----
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
-----
```

#### 6.4.3 Pertanyaan

- ```
void insertionSort(){
    for (int i = 0; i < listMhs.length; i++){
        Mahasiswa16 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j-1].ipk < temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Hasil

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

## 6.5 Latihan

### Dosen16

```
public class Dosen16 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    Dosen16(){

    }

    Dosen16(String kd, String name, boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

    public void tampil() {
        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 + " tahun");
    }

    int Usia() {
        return usia;
    }
}
```

## DataDosen16

```
public class DataDosen16 {
    Dosen16 [] dataDosen = new Dosen16[10];
    int idx;

    void tambah (Dosen16 dsn){
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        }else {
            System.out.println("Data sudah penuh");
        }
    }

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

    void sortingASC(){
        for (int i = 0; i < idx-1; i++){
            for (int j = 0; j < idx-1-i; j++){
                if (dataDosen[j].usia > dataDosen[j+1].usia) {
                    Dosen16 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j+1];
                    dataDosen[j+1] = temp;
                }
            }
        }
    }

    void SortingDSC(){
        for (int i = 0; i < idx-1; i++){
            int idxMax = i;
            for (int j = i+1; j < idx; j++){
                if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                    idxMax = j;
                }
            }
            Dosen16 temp = dataDosen[idxMax];
            dataDosen[idxMax] = dataDosen[i];
            dataDosen[i] = temp;
        }
    }
}
```

## DosenMain

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

        DataDosen16 data = new DataDosen16();
        int pilihan;

        do {
            System.out.println("Menu:");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampil Data Dosen");
            System.out.println("3. Sorting ASC (dari termuda ke tertua)");
            System.out.println("4. Sorting DSC (dari tertua ke termuda)");
            System.out.println("5. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Kode          : ");
                    String kode = sc.nextLine();
                    System.out.print("Nama          : ");
                    String nama = sc.nextLine();
                    System.out.print("Jenis Kelamin: ");
                    boolean jenisKelamin = sc.nextBoolean();
                    System.out.print("Usia          : ");
                    int usia = sc.nextInt();
                    sc.nextLine();

                    Dosen16 dosen = new Dosen16(kode, nama, jenisKelamin, usia);
                    data.tambah(dosen);
                    break;

                case 2:
                    data.tampil();
                    break;

                case 3:
                    data.sortingASC();
                    System.out.println("Sorting dari termuda ke tertua");
                    data.tampil();
                    break;

                case 4:
                    data.SortingDSC();
                    System.out.println("Sorting dari tertua ke termuda");
                    data.tampil();
                    break;

                case 5:
                    System.out.println("Keluar dari program");
                    break;
            }
            System.out.println();
        } while (pilihan != 5);
    }
}
```



## Hasil

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
Pilih menu: 1
Kode       : 123
Nama       : Pak Fauzi
Jenis Kelamin: true
Usia       : 30
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
Pilih menu: 1
Kode       : 124
Nama       : Pak Agus
Jenis Kelamin: true
Usia       : 35
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
Pilih menu: 1
Kode       : 120
Nama       : Bu Tea
Jenis Kelamin: false
Usia       : 28
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
```

```
Pilih menu: 2
Kode Dosen : 123
Nama Dosen : Pak Fauzi
Jenis Kelamin: Laki-laki
Usia       : 30 tahun
```

```
-----
Kode Dosen : 124
Nama Dosen : Pak Agus
Jenis Kelamin: Laki-laki
Usia       : 35 tahun
```

```
-----
Kode Dosen : 120
Nama Dosen : Bu Tea
Jenis Kelamin: Perempuan
Usia       : 28 tahun
-----
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
Pilih menu: 3
Sorting dari termuda ke tertua
Kode Dosen : 120
Nama Dosen : Bu Tea
Jenis Kelamin: Perempuan
Usia       : 28 tahun
```

```
-----
Kode Dosen : 123
Nama Dosen : Pak Fauzi
Jenis Kelamin: Laki-laki
Usia       : 30 tahun
```

```
-----
Kode Dosen : 124
Nama Dosen : Pak Agus
Jenis Kelamin: Laki-laki
Usia       : 35 tahun
-----
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
Pilih menu: 4
Sorting dari tertua ke termuda
Kode Dosen : 124
Nama Dosen : Pak Agus
Jenis Kelamin: Laki-laki
Usia       : 35 tahun
```

```
-----
Kode Dosen : 123
Nama Dosen : Pak Fauzi
Jenis Kelamin: Laki-laki
Usia       : 30 tahun
```

```
-----
Kode Dosen : 120
Nama Dosen : Bu Tea
Jenis Kelamin: Perempuan
Usia       : 28 tahun
-----
```

```
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (dari termuda ke tertua)
4. Sorting DSC (dari tertua ke termuda)
5. Keluar
Pilih menu: 5
Keluar dari program
```