

Laporan Hasil Pratikum
Algoritma Dan Struktur Data

Jobsheet 7



Nama : Zacky Rio Orlando

NIM : 244107020086

Kelas : 1E

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Praktikum

2025

7.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

Class Mahasiswa27

```
public class Mahasiswa27 {  
    String nim;  
    String nama;  
    String kelas;  
    double ipk;  
  
    Mahasiswa27() {  
    }  
  
    Mahasiswa27(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);  
    }  
}
```

Class MahasiswaBerprestasi27

```
public class MahasiswaBerprestasi27 {
    Mahasiswa27 [] listMhs = new Mahasiswa27[5];
    int idx;

    void tambah (Mahasiswa27 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil () {
        for (Mahasiswa27 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) {
                    Mahasiswa27 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa27 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }

    void insertionSort() {
        for (int i=1; i<listMhs.length; i++) {
            Mahasiswa27 temp = listMhs[i];
            int j=i;
            while (j>0 && listMhs[j-1].ipk<temp.ipk) {
                listMhs[j] = listMhs[j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}
```

```

int sequentialSearching(double cari) {
    int posisi = -1;
    for (int j=0; j<listMhs.length; j++) {
        if (listMhs[j].ipk == cari) {
            posisi = j;
            break;
        }
    }
    return posisi;
}

void tampilPosisi (double x, int pos) {
    if (pos!= -1) {
        System.out.println("data mahasiswa dengan IPK : " +x+ "
ditemukan pada indeks " + pos );
    }
    else {
        System.out.println("data " +x+ "tidak ditemukan");
    }
}

void tampilDataSearch(double x, int pos) {
    if (pos != -1) {
        System.out.println("nim\t : " +listMhs[pos].nim);
        System.out.println("nama\t : " +listMhs[pos].nama);
        System.out.println("kelas\t : " +listMhs[pos].kelas);
        System.out.println("ipk\t : " +x);
    }
    else {
        System.out.println("Data mahasiswa dengan IPK " +x+ " tidak
ditemukan");
    }
}

```

Class MahasiswaDemo27

```

import java.util.Scanner;
public class MahasiswaDemo27 {
    public static void main(String[] args) {
        MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27();
        Scanner sc = new Scanner(System.in);
        int jumMhs = 5;

        for (int i=0; i<jumMhs; i++) {
            System.out.println("Masukkan Data 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   : ");
            String ipk = sc.nextLine();
            Double ipk = Double.parseDouble(ipk);
            System.out.println("-----");
            list.tambah (new Mahasiswa27(nim, nama, kelas, ipk));
        }

        list.tampil();
        //melakukan pencarian data sequential
        System.out.println("-----");
    };

    System.out.println("Pencarian data");
    System.out.println("-----");
    ";

    System.out.println("masukkan ipk mahasiswa yang dicari: ");
    System.out.print("IPK: ");
    double cari = sc.nextDouble();
    sc.nextLine();

    System.out.println("menggunakan sequential searching");
    double posisi = list.sequentialSearching(cari);
    int pss = (int)posisi;
    list.tampilPosisi(cari, pss);
    list.tampilDataSearch(cari, pss);
}
}

```

Hasil Output dari kode program diatas

Masukkan Data Mahasiswa ke-1

NIM : 111

Nama : adi

Kelas : 2

IPK : 3.6

Masukkan Data Mahasiswa ke-2

NIM : 222

Nama : tio

Kelas : 2

IPK : 3.8

Masukkan Data Mahasiswa ke-3

NIM : 333

Nama : lia

Kelas : 2

IPK : 3.0

Masukkan Data Mahasiswa ke-4

NIM : 444

Nama : lia

Kelas : 2

IPK : 3.5

Masukkan Data Mahasiswa ke-5

NIM : 555

Nama : fia

Kelas : 2

IPK : 3.3

```

Nama: adi
NIM: 111
Kelas: 2
IPK: 3.6
-----
Nama: tio
NIM: 222
Kelas: 2
IPK: 3.8
-----
Nama: lia
NIM: 333
Kelas: 2
IPK: 3.0
-----
Nama: lia
NIM: 444
Kelas: 2
IPK: 3.5
-----
Nama: fia
NIM: 555
Kelas: 2
IPK: 3.3
-----

```

```

-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,5
menggunakan sequential searching
data mahasiswa dengan IPK :3.5 ditemukan pada indeks 3
nim      : 444
nama     : lia
kelas    : 2
ipk      : 3.5

```

7.2.3. Pertanyaan

1. Jelaskan perbedaan metod **tampilDataSearch** dan **tampilPosisi** pada class MahasiswaBerprestasi!

Jawaban :

Metod tampilPosisi() dan tampilDataSearch() pada kelas MahasiswaBerprestasi²⁷ digunakan setelah mencari mahasiswa berdasarkan IPK. tampilPosisi() berfungsi untuk menunjukkan di urutan ke berapa mahasiswa itu ditemukan dalam daftar, sedangkan tampilDataSearch() menampilkan data lengkap mahasiswa itu, seperti NIM, nama, kelas, dan IPK. Jadi, tampilPosisi() hanya menunjukkan letaknya, dan tampilDataSearch() menampilkan isi datanya.

Keduanya saling melengkapi agar kita tahu siapa mahasiswa yang dicari dan di posisi mana dia berada.

2. Jelaskan fungsi **break** pada kode program dibawah ini!

```
if (listMhs[j].ipk==cari){  
    posisi=j;  
    break;  
}
```

Jawaban :

Fungsi break pada kode tersebut digunakan untuk menghentikan proses pencarian ketika data yang dicari sudah ketemu. Jadi, saat IPK mahasiswa yang dicari sama dengan IPK dalam data (listMhs[j].ipk == cari), maka program akan menyimpan posisi data tersebut, lalu langsung keluar dari perulangan. Dengan begitu, program tidak lanjut mengecek data mahasiswa lainnya karena yang dicari sudah ditemukan. Ini membuat pencarian jadi lebih cepat dan efisien.

7.3 Searching / Pencarian Menggunakan Binary Search

Class Mahasiswa27

```
public class Mahasiswa27 {  
    String nim;  
    String nama;  
    String kelas;  
    double ipk;  
  
    Mahasiswa27() {  
    }  
  
    Mahasiswa27(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);  
    }  
}
```


Class MahasiswaBerprestasi27

```
public class MahasiswaBerprestasi27 {
    Mahasiswa27 [] listMhs = new Mahasiswa27[5];
    int idx;

    void tambah (Mahasiswa27 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil () {
        for (Mahasiswa27 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) {
                    Mahasiswa27 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa27 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }

    void insertionSort() {
        for (int i=1; i<listMhs.length; i++) {
            Mahasiswa27 temp = listMhs[i];
            int j=i;
            while (j>0 && listMhs[j-1].ipk<temp.ipk) {
                listMhs[j] = listMhs[j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}
```

```

int sequentialSearching(double cari) {
    int posisi = -1;
    for (int j=0; j<listMhs.length; j++) {
        if (listMhs[j].ipk == cari) {
            posisi = j;
            break;
        }
    }
    return posisi;
}

void tampilPosisi (double x, int pos) {
    if (pos!= -1) {
        System.out.println("data mahasiswa dengan IPK :"+x+ "
ditemukan pada indeks " + pos );
    }
    else {
        System.out.println("data "+x+ "tidak ditemukan");
    }
}

void tampilDataSearch(double x, int pos) {
    if (pos != -1) {
        System.out.println("nim\t : " +listMhs[pos].nim);
        System.out.println("nama\t : " +listMhs[pos].nama);
        System.out.println("kelas\t : " +listMhs[pos].kelas);
        System.out.println("ipk\t : " +x);
    }
    else {
        System.out.println("Data mahasiswa dengan IPK "+x+ " tidak
ditemukan");
    }
}

int findBinarySearch(double cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (left+right)/2;
        if (cari == listMhs[mid].ipk) {
            return (mid);
        }
        else if (listMhs[mid].ipk > cari) {
            return findBinarySearch(cari, left, mid-1);
        }
        else {
            return findBinarySearch(cari, mid+1, right);
        }
    }
    return -1;
}
}

```

Class MahasiswaDemo27

```
import java.util.Scanner;
public class MahasiswaDemo27 {
    public static void main(String[] args) {
        MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27();
        Scanner sc = new Scanner(System.in);
        int jumMhs = 5;

        for (int i=0; i<jumMhs; i++) {
            System.out.println("Masukkan Data 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   : ");
            String ip = sc.nextLine();
            Double ipk = Double.parseDouble(ip);
            System.out.println("-----");
            list.tambah (new Mahasiswa27(nim, nama, kelas, ipk));
        }

        list.tampil();
        //melakukan pencarian data Binary
        System.out.println("-----");
        System.out.println("Pencarian data");
        System.out.println("-----");
        System.out.println("masukkan ipk mahasiswa yang dicari: ");
        System.out.print("IPK: ");
        double cari = sc.nextDouble();
        System.out.println("-----");
        System.out.println("menggunakan binary search");
        System.out.println("-----");
        double posisi2 = list.findBinarySearch(cari, 0, jumMhs-1);
        int pss2 = (int)posisi2;
        list.tampilPosisi(cari, pss2);
        list.tampilDataSearch(cari, pss2);
    }
}
```

Hasil Output dari kode program diatas

Masukkan Data Mahasiswa ke-1

NIM : 111

Nama : adi

Kelas : 2

IPK : 3.1

Masukkan Data Mahasiswa ke-2

NIM : 222

Nama : ila

Kelas : 2

IPK : 3.2

Masukkan Data Mahasiswa ke-3

NIM : 333

Nama : lia

Kelas : 2

IPK : 3.3

Masukkan Data Mahasiswa ke-4

NIM : 444

Nama : susi

Kelas : 2

IPK : 3.5

Masukkan Data Mahasiswa ke-5

NIM : 555

Nama : anita

Kelas : 2

IPK : 3.7

```
Nama: adi
NIM: 111
Kelas: 2
IPK: 3.1
-----
Nama: ila
NIM: 222
Kelas: 2
IPK: 3.2
-----
Nama: lia
NIM: 333
Kelas: 2
IPK: 3.3
-----
Nama: susi
NIM: 444
Kelas: 2
IPK: 3.5
-----
Nama: anita
NIM: 555
Kelas: 2
IPK: 3.7
-----
```

```
-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,7
-----
menggunakan binary search
-----
data mahasiswa dengan IPK :3.7 ditemukan pada indeks 4
nim      : 555
nama     : anita
kelas    : 2
ipk      : 3.7
```

7.3.3. Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!

```
mid = (left+right)/2;
```

2. Tunjukkan pada kode program yang mana proses conquer dijalankan!

```
else if (listMhs[mid].ipk > cari) {  
    return findBinarySearch(cari, left, mid-1);  
}  
else {  
    return findBinarySearch(cari, mid+1, right);  
}
```

3. Jika data IPK yang dimasukkan tidakurut. Apakah program masih dapat berjalan? Mengapa demikian!

Jawaban :

Program masih dapat berjalan meskipun data IPK yang dimasukkan tidakurut, karena secara teknis tidak ada kesalahan sintaks yang menghentikan program. Namun, hasil pencarian menggunakan binary search bisa tidak akurat atau gagal menemukan data yang sebenarnya ada. Hal ini disebabkan karena binary search hanya bekerja dengan benar jika data sudah terurut, karena algoritma ini membagi data dan memutuskan arah pencarian berdasarkan urutan. Jika data tidakurut, pembagian dan arah pencarian bisa salah, sehingga hasilnya tidak sesuai. Maka dari itu, sangat penting untuk mengurutkan data terlebih dahulu sebelum melakukan binary search.

4. Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (missal : 3.8, 3.7, 3.5, 3.4, 3.2) dan elemen yang dicari adalah 3.2. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai

Jawaban :

Jika IPK dimasukkan dari yang paling besar ke yang paling kecil (misalnya: 3.8, 3.7, 3.5, 3.4, 3.2) lalu dilakukan pencarian menggunakan binary search seperti biasa, maka hasilnya tidak akan sesuai. Ini karena binary search hanya bisa bekerja dengan benar jika data disusun dari kecil ke besar. Kalau data diurutkan dari besar ke kecil, maka arah pencarian dalam binary search jadi terbalik dan bisa menyebabkan data tidak ditemukan, padahal sebenarnya ada. Supaya hasilnya benar, kamu bisa mengurutkan data dari kecil ke besar dulu, atau ubah cara kerja binary search-nya supaya cocok dengan urutan dari besar ke kecil.

5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

Jawaban :

Class Mahasiswa27

```
public class Mahasiswa27 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa27() {
    }

    Mahasiswa27(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);
    }
}
```

class MahasiswaBerprestasi27

```

public class MahasiswaBerprestasi27 {
    Mahasiswa27 [] listMhs = new Mahasiswa27[5];
    int idx;

    void tambah (Mahasiswa27 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil () {
        for (Mahasiswa27 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) {
                    Mahasiswa27 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa27 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }

    void insertionSort() {
        for (int i=1; i<listMhs.length; i++) {
            Mahasiswa27 temp = listMhs[i];
            int j=i;
            while (j>0 && listMhs[j-1].ipk<temp.ipk) {
                listMhs[j] = listMhs[j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}

```



```

int sequentialSearching(double cari) {
    int posisi = -1;
    for (int j=0; j<listMhs.length; j++) {
        if (listMhs[j].ipk == cari) {
            posisi = j;
            break;
        }
    }
    return posisi;
}

void tampilPosisi (double x, int pos) {
    if (pos!= -1) {
        System.out.println("data mahasiswa dengan IPK : " +x+ "
ditemukan pada indeks " + pos );
    }
    else {
        System.out.println("data " +x+ "tidak ditemukan");
    }
}

void tampilDataSearch(double x, int pos) {
    if (pos != -1) {
        System.out.println("nim\t : " +listMhs[pos].nim);
        System.out.println("nama\t : " +listMhs[pos].nama);
        System.out.println("kelas\t : " +listMhs[pos].kelas);
        System.out.println("ipk\t : " +x);
    }
    else {
        System.out.println("Data mahasiswa dengan IPK " +x+ " tidak
ditemukan");
    }
}

int findBinarySearch(double cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (left+right)/2;
        if (cari == listMhs[mid].ipk) {
            return (mid);
        }
        else if (listMhs[mid].ipk > cari) {
            return findBinarySearch(cari, left, mid-1);
        }
        else {
            return findBinarySearch(cari, mid+1, right);
        }
    }
    return -1;
}

MahasiswaBerprestasi27(int jumlah) {
    listMhs = new Mahasiswa27[jumlah];
    idx = 0;
}
}

```

class MahasiswaDemo27

```
import java.util.Scanner;
public class MahasiswaDemo27 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan jumlah mahasiswa: ");
        int jumMhs = sc.nextInt();
        sc.nextLine();

        MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27(jumMhs);

        for (int i=0; i<jumMhs; i++) {
            System.out.println("Masukkan Data 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   : ");
            String ipk = sc.nextLine();
            Double ipk = Double.parseDouble(ipk);
            System.out.println("-----");
            list.tambah (new Mahasiswa27(nim, nama, kelas, ipk));
        }

        list.tampil();
        //melakukan pencarian data Binary
        System.out.println("-----");
        System.out.println("Pencarian data");
        System.out.println("-----");
        System.out.println("masukkan ipk mahasiswa yang dicari: ");
        System.out.print("IPK: ");
        double cari = sc.nextDouble();
        System.out.println("-----");
        System.out.println("menggunakan binary search");
        System.out.println("-----");
        double posisi2 = list.findBinarySearch(cari, 0, jumMhs-1);
        int pss2 = (int)posisi2;
        list.tampilPosisi(cari, pss2);
        list.tampilDataSearch(cari, pss2);
    }
}
```

Hasil Output dari kode program diatas

```

Masukkan jumlah mahasiswa: 2
Masukkan Data Mahasiswa ke-1
NIM   : 111
Nama  : adi
Kelas : 2
IPK   : 3.1
-----
Masukkan Data Mahasiswa ke-2
NIM   : 222
Nama  : ila
Kelas : 2
IPK   : 3.2
-----
Nama: adi
NIM: 111
Kelas: 2
IPK: 3.1
-----
Nama: ila
NIM: 222
Kelas: 2
IPK: 3.2
-----

```

```

-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,1
-----
menggunakan binary search
-----
data mahasiswa dengan IPK :3.1 ditemukan pada indeks 0
nim      : 111
nama     : adi
kelas    : 2
ipk      : 3.1

```

7.5. Latihan Praktikum

1. Pada Latihan praktikum pertemuan sebelumnya pada Jobsheet 6 yang terdapat 3 class yaitu **Dosen**<no presensi>, **DataDosen**<no presensi> , dan **DosenDemo**<no presensi>, tambahkan method:

- PencarianDataSequential<no presensi> : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.
- PencarianDataBinary<no presensi> : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.

c. Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

Jawaban

Class Dosen27

```
public class Dosen27 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    Dosen27() {
    }

    Dosen27(String kd, String name, boolean jk, int age) {
        kode = kd;
        nama = name;
        usia = age;
        this.jenisKelamin = jk;
    }

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

Class DataDosen27

```
public class DataDosen27 {
    Dosen27[] dataDosen27 = new Dosen27[10];
    int idx = 0;

    void tambah(Dosen27 dsn) {
        if (idx < dataDosen27.length) {
            dataDosen27[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data penuh!");
        }
    }

    void tampil() {
        for (int i = 0; i < idx; i++) {
            dataDosen27[i].tampil();
        }
    }

    void sortingASC() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - i - 1; j++) {
                if (dataDosen27[j].usia > dataDosen27[j + 1].usia) {
                    Dosen27 temp = dataDosen27[j];
                    dataDosen27[j] = dataDosen27[j + 1];
                    dataDosen27[j + 1] = temp;
                }
            }
        }
    }

    void sortingDSC() {
        for (int i = 0; i < idx - 1; i++) {
            int maxIdx = i;
            for (int j = i + 1; j < idx; j++) {
                if (dataDosen27[j].usia > dataDosen27[maxIdx].usia) {
                    maxIdx = j;
                }
            }
            Dosen27 temp = dataDosen27[maxIdx];
            dataDosen27[maxIdx] = dataDosen27[i];
            dataDosen27[i] = temp;
        }
    }

    void pencarianDataSequential27(String namaCari) {
        int jumlahTemuan = 0;
        for (int i = 0; i < idx; i++) {
            if (dataDosen27[i].nama.equalsIgnoreCase(namaCari)) {
                dataDosen27[i].tampil();
                jumlahTemuan++;
            }
        }
        if (jumlahTemuan == 0) {
            System.out.println("Data dosen dengan nama \"" + namaCari + "\" tidak ditemukan.");
        } else if (jumlahTemuan > 1) {
            System.out.println("Peringatan: Ditemukan lebih dari satu dosen dengan nama \"" + namaCari + "\".");
        }
    }
}
```

```

void pencarianDataBinary27(int usiaCari) {
    sortingASC();
    int left = 0;
    int right = idx - 1;
    int jumlahTemuan = 0;

    while (left <= right) {
        int mid = (left + right) / 2;
        if (dataDosen27[mid].usia == usiaCari) {
            int i = mid;
            while (i >= 0 && dataDosen27[i].usia == usiaCari) {
                dataDosen27[i].tampil();
                jumlahTemuan++;
                i--;
            }
            i = mid + 1;
            while (i < idx && dataDosen27[i].usia == usiaCari) {
                dataDosen27[i].tampil();
                jumlahTemuan++;
                i++;
            }
            break;
        } else if (dataDosen27[mid].usia > usiaCari) {
            right = mid - 1;
        } else {
            left = mid + 1;
        }
    }

    if (jumlahTemuan == 0) {
        System.out.println("Data dosen dengan usia " + usiaCari + "
tidak ditemukan.");
    } else if (jumlahTemuan > 1) {
        System.out.println("Peringatan: Ditemukan lebih dari satu dosen
dengan usia " + usiaCari + ".");
    }
}
}

```

Class DosenDemo27

```
import java.util.Scanner;

public class DosenDemo27 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen27 data27 = new DataDosen27();
        int pilihan;

        do {
            System.out.println("\nMENU");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Sorting Usia (ASC)");
            System.out.println("4. Sorting Usia (DSC)");
            System.out.println("5. Cari Nama Dosen (Sequential)");
            System.out.println("6. Cari Usia Dosen (Binary)");
            System.out.println("7. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Masukkan Kode Dosen: ");
                    String kode = sc.nextLine();
                    System.out.print("Masukkan Nama Dosen: ");
                    String nama = sc.nextLine();
                    System.out.print("Masukkan Jenis Kelamin: ");
                    String inputJK = sc.nextLine();
                    boolean jk = inputJK.equalsIgnoreCase("laki-laki");
                    System.out.print("Masukkan Usia: ");
                    int usia = sc.nextInt();
                    sc.nextLine();
                    data27.tambah(new Dosen27(kode, nama, jk, usia));
                    break;
                case 2:
                    data27.tampil();
                    break;
                case 3:
                    data27.sortingASC();
                    System.out.println("Data telah diurutkan dari termuda ke
                    tertua.");
                    break;
                case 4:
                    data27.sortingDSC();
                    System.out.println("Data telah diurutkan dari tertua ke
                    termuda.");
                    break;
                case 5:
                    System.out.print("Masukkan nama yang dicari: ");
                    String namaCari = sc.nextLine();
                    data27.pencarianDataSequential27(namaCari);
                    break;
                case 6:
                    System.out.print("Masukkan usia yang dicari: ");
                    int usiaCari = sc.nextInt();
                    data27.pencarianDataBinary27(usiaCari);
                    break;
                case 7:
                    System.out.println("Keluar dari program.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid!");
            }
        } while (pilihan != 7);
    }
}
```

Hasil Output dari kode program diatas

```
MENU
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting Usia (ASC)
4. Sorting Usia (DSC)
5. Cari Nama Dosen (Sequential)
6. Cari Usia Dosen (Binary)
7. Keluar
Pilih menu: 1
Masukkan Kode Dosen: ALSD
Masukkan Nama Dosen: Vivi Nur Wijayani
ngrum, S.Kom., M.Kom.
Masukkan Jenis Kelamin: Perempuan
Masukkan Usia: 30
```

```
MENU
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting Usia (ASC)
4. Sorting Usia (DSC)
5. Cari Nama Dosen (Sequential)
6. Cari Usia Dosen (Binary)
7. Keluar
Pilih menu: 1
Masukkan Kode Dosen: SISOP
Masukkan Nama Dosen: Samsul Arifin, S.
Kom., MMSI.
Masukkan Jenis Kelamin: Laki-laki
Masukkan Usia: 35
```

```
MENU
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting Usia (ASC)
4. Sorting Usia (DSC)
5. Cari Nama Dosen (Sequential)
6. Cari Usia Dosen (Binary)
7. Keluar
Pilih menu: 5
Masukkan nama yang dicari: Vivi Nur Wijayaningrum, S.Kom., M.Kom.
Kode: ALSD
Nama: Vivi Nur Wijayaningrum, S.Kom., M.Kom.
Jenis Kelamin: Perempuan
Usia: 30
-----
```


MENU

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting Usia (ASC)
4. Sorting Usia (DSC)
5. Cari Nama Dosen (Sequential)
6. Cari Usia Dosen (Binary)
7. Keluar

Pilih menu: 6

Masukkan usia yang dicari: 35

Kode: SISOP

Nama: Samsul Arifin, S.Kom., MMSI.

Jenis Kelamin: Laki-laki

Usia: 35
