

**JOBSHEET 7**  
**LAPORAN HASIL PRAKTIKUM**  
**ALGORITMA DAN STRUKTUR**  
**DATA**



**MUHAMMAD DAFFI FIROS ZAIDAN**

**244107020182**

**TI 1E**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2025**

## JOBSHEET VII SEARCHING

### 7.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

#### 1. Kode Program

Mahasiswa Berprestasi17

```
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");
    }
}
}
```

MahasiswaDemo17

```
import java.util.Scanner;
public class MahasiswaDemo {
    public static void main(String[] args) {
        MahasiswaBerprestasi17 list = new MahasiswaBerprestasi17();
        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\t: ");
            String nim = sc.nextLine();
            System.out.print("Nama\t: ");
            String nama = sc.nextLine();
            System.out.print("Kelas\t: ");
            String kelas = sc.nextLine();
            System.out.print("IPK\t: ");
            Double ipk = Double.parseDouble(sc.nextLine());
            list.tambah(new Mahasiswa17(nim, nama, kelas, ipk));
        }
    }
}
```

```
list.tampil();

        System.out.println("=====");
        System.out.println("== Pencarian Data ==");
        System.out.println("=====");
        System.out.print("Masukkan IPK mahasiswa yang dicari: ");
        double cari = sc.nextDouble();

        System.out.println("== Menggunakan sequential searching ==");
        int posisi = list.sequentialSearching(cari);
        list.tampilPosisi(cari, posisi);
        list.tampilDataSearch(cari, posisi);
```

## 2. Hasil Kode Program

```
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     : ila
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 : ila
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
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD> █
```

### 7.2.3. Pertanyaan

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

Jawab:

Metode **tampilDataSearch** digunakan untuk menampilkan hasil pencarian berdasarkan kriteria tertentu, seperti IPK yang diminta oleh user. contoh, metode ini menunjukkan data mahasiswa yang ditemukan sesuai dengan pencarian IPK, mencantumkan informasi relevan seperti NIM, nama, dan kelas.

Sementara itu, metode **tampilPosisi** digunakan untuk menampilkan posisi atau indeks dari mahasiswa yang ditemukan di dalam daftar. Ini akan memberikan informasi tentang lokasi mahasiswa dalam array atau list berdasarkan pencarian yang telah dilakukan.

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

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

Tujuan Penggunaan Kode Diatas adalah Menghentikan perulangan lebih awal ketika kondisi listMhs[j].ip telah ditemukan.

### 7.3 Searching / Pencarian Menggunakan Binary Search

1. Kode Program

MahasiswaBerprestasi

```
int findBinarySearch(double cari, int left, int right) {  
    int mid;  
    if (right >= left) {  
        mid = left + (right - left) / 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;  
}
```

MahasiswaDemo

```
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, list.listMhs.length - 1);  
int pss2 = (int) posisi2;  
list.tampilPosisi(cari, pss2);  
list.tampilDataSearch(cari, pss2);  
}  
}
```

## 2. Hasil Kode Program

```
Masukkan Data Mahasiswa ke-1
NIM      : 111
Nama     : adi
Kelas   : 2
IPK      : 3.1
Masukkan Data Mahasiswa ke-2
NIM      : 222
Nama     : lia
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 : lia
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
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
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD>
```

### 7.3.3. Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan  
 $mid = left + (right - left) / 2;$
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!

```
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);  
}
```

3. Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!  
Tidak, Binary search hanya bisa bekerja pada jika data yang ingin dicari sudah urut jika data belum urut maka harus diurutkan terlebih dahulu
4. Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (misal : 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 search agar hasilnya sesuai

```
int findBinarySearch(double cari, int left, int right) {  
    int mid;  
    if (right >= left) {  
        mid = left + (right - left) / 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;  
}
```

Mengubah tanda > menjadi < pada kondisi else if

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

MahasiswaBerprestasi17

```
public class MahasiswaBerprestasi17 {  
    Mahasiswa17[] listMhs;  
    int idx;  
  
    public MahasiswaBerprestasi17(int jumlah) {  
        listMhs = new Mahasiswa17[jumlah];  
        idx = 0;  
    }  
}
```

```
void tampil() {  
    for (Mahasiswa17 m : listMhs) {  
        if (m != null) {  
            System.out.println("-----");  
            m.tampilkanInformasi();  
        }  
    }  
}
```

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Masukkan jumlah mahasiswa: ");
    int jumMhs = sc.nextInt();
    sc.nextLine();
    MahasiswaBerprestasi17 list = new MahasiswaBerprestasi17(jumMhs);
}
```

### 7.5. Latihan Praktikum

1. Pada Latihan praktikum pertemuan sebelumnya pada Jobsheet 6 yang terdapat 3 class yaitu Dosen, DataDosen, dan DosenDemo, tambahkan method:
  - a. PencarianDataSequential : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.
  - b. PencarianDataBinary : 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!

- Kode Program  
Dosen17

```
public class Dosen17 {
    String nama;
    int usia;
    String nip;
    String jenisKelamin;

    public Dosen17(String nama, int usia, String nip, String jenisKelamin) {
        this.nama = nama;
        this.usia = usia;
        this.nip = nip;
        this.jenisKelamin = jenisKelamin;
    }
}
```

### Data Dosen

```
public class DataDosen17 {
    Dosen17[] dosen;
    int jml;

    public DataDosen17(int jumlah) {
        dosen = new Dosen17[jumlah];
        jml = 0;
    }

    public void tambah(Dosen17 d) {
        if(jml < dosen.length) {
            dosen[jml] = d;
            jml++;
        } else {
            System.out.println("Data sudah penuh!");
        }
    }
}
```

```

public void tampil() {
    int jmlLaki = 0, jmlPerempuan = 0;

    for(int i = 0; i < jml; i++) {
        System.out.println("\nData Dosen ke-" + (i+1));
        System.out.println("Nama: " + dosen[i].nama);
        System.out.println("Usia: " + dosen[i].usia);
        System.out.println("NIP: " + dosen[i].nip);
        System.out.println("Jenis Kelamin: " + dosen[i].jenisKelamin);
        System.out.println("-----");

        if(dosen[i].jenisKelamin.equalsIgnoreCase("L")) {
            jmlLaki++;
        } else if(dosen[i].jenisKelamin.equalsIgnoreCase("P")) {
            jmlPerempuan++;
        }
    }

    System.out.println("\nJumlah Dosen Berdasarkan Jenis Kelamin:");
    System.out.println("Laki-laki: " + jmlLaki + " orang");
    System.out.println("Perempuan: " + jmlPerempuan + " orang");
}

public void tampilkanInfo() {
    for(int i = 0; i < jml; i++) {
        System.out.println("Data Dosen ke-" + (i+1));
        System.out.println("Nama: " + dosen[i].nama);
        System.out.println("Usia: " + dosen[i].usia);
        System.out.println("NIP: " + dosen[i].nip);
        System.out.println("-----");
    }
}

public void jumlahDosenPerJenisKelamin() {
    int jmlLaki = 0, jmlPerempuan = 0;
    for(int i = 0; i < jml; i++) {
        if(dosen[i].jenisKelamin.equalsIgnoreCase("L")) {
            jmlLaki++;
        } else if(dosen[i].jenisKelamin.equalsIgnoreCase("P")) {
            jmlPerempuan++;
        }
    }
    System.out.println("Jumlah dosen laki-laki: " + jmlLaki);
    System.out.println("Jumlah dosen perempuan: " + jmlPerempuan);
}

public void pencarianDataSequential(String nama) {
    boolean ditemukan = false;
    int jumlahDitemukan = 0;

    System.out.println("\nHasil Pencarian untuk nama: " + nama);
    System.out.println("=====");

    for(int i = 0; i < jml; i++) {
        if(dosen[i].nama.equalsIgnoreCase(nama)) {
            if (jumlahDitemukan == 0) {
                System.out.println("Data ditemukan pada indeks berikut:");
            }
            jumlahDitemukan++;
            System.out.println("\nData ke-" + jumlahDitemukan + " pada indeks " + i);
            System.out.println("Nama: " + dosen[i].nama);
            System.out.println("Usia: " + dosen[i].usia);
            System.out.println("NIP: " + dosen[i].nip);
            System.out.println("Jenis Kelamin: " + dosen[i].jenisKelamin);
            ditemukan = true;
        }
    }

    if(!ditemukan) {
        System.out.println("Data " + nama + " tidak ditemukan!");
    } else if(jumlahDitemukan > 1) {
        System.out.println("\n[PERINGATAN] Ditemukan " + jumlahDitemukan + " data dengan nama yang sama!");
        System.out.println("Silakan gunakan informasi tambahan (seperti NIP) untuk identifikasi lebih spesifik.");
    }
}

public void sortinqAsc() {

```





```

public void sortingAsc() {
    for(int i = 0; i < jml-1; i++) {
        for(int j = 0; j < jml-i-1; j++) {
            if(dosen[j].usia > dosen[j+1].usia) {

                Dosen17 tmp = dosen[j];
                dosen[j] = dosen[j+1];
                dosen[j+1] = tmp;

            }
        }
    }
    System.out.println("Data telah diurutkan berdasarkan usia secara ascending!");
}

public void insertionSort() {
    for(int i = 1; i < jml; i++) {
        Dosen17 temp = dosen[i];
        int j = i;
        while(j > 0 && dosen[j-1].usia > temp.usia) {
            dosen[j] = dosen[j-1];
            j--;
        }
        dosen[j] = temp;
    }
}

public void pencarianDataBinary(int usia) {
    insertionSort();
    boolean ditemukan = false;
    int jumlahDitemukan = 0;
    int awal = 0, akhir = jml - 1;

    System.out.println("\nHasil Pencarian untuk usia: " + usia);
    System.out.println("=====");

    while(awal <= akhir) {
        int tengah = (awal + akhir) / 2;
        if(dosen[tengah].usia == usia) {
            jumlahDitemukan++;
            if (jumlahDitemukan == 1) {
                System.out.println("Data ditemukan pada indeks berikut:");
            }

            System.out.println("\nData ke-" + jumlahDitemukan + " pada indeks " + tengah);
            System.out.println("Nama: " + dosen[tengah].nama);
            System.out.println("Usia: " + dosen[tengah].usia);
            System.out.println("NIP: " + dosen[tengah].nip);
            System.out.println("Jenis Kelamin: " + dosen[tengah].jenisKelamin);
            ditemukan = true;

            int temp = tengah - 1;
            while(temp >= 0 && dosen[temp].usia == usia) {
                jumlahDitemukan++;
                System.out.println("\nData ke-" + jumlahDitemukan + " pada indeks " + temp);
                System.out.println("Nama: " + dosen[temp].nama);
                System.out.println("Usia: " + dosen[temp].usia);
                System.out.println("NIP: " + dosen[temp].nip);
                System.out.println("Jenis Kelamin: " + dosen[temp].jenisKelamin);
                temp--;
            }

            temp = tengah + 1;
            while(temp < jml && dosen[temp].usia == usia) {
                jumlahDitemukan++;
                System.out.println("\nData ke-" + jumlahDitemukan + " pada indeks " + temp);
                System.out.println("Nama: " + dosen[temp].nama);
                System.out.println("Usia: " + dosen[temp].usia);
                System.out.println("NIP: " + dosen[temp].nip);
                System.out.println("Jenis Kelamin: " + dosen[temp].jenisKelamin);
                temp++;
            }

            if(jumlahDitemukan > 1) {
                System.out.println("\n[PERINGATAN] Ditemukan " + jumlahDitemukan + " data
dengan usia yang sama!");
            }
            break;
        } else if(dosen[tengah].usia > usia) {
            akhir = tengah - 1;
        } else {
            awal = tengah + 1;
        }
    }
}

```

```

if(jumlahDitemukan > 1) {
    System.out.println("\n[PERINGATAN] Ditemukan " + jumlahDitemukan + " data dengan
usia yang sama!");
}
break;
} else if(dosen[tengah].usia > usia) {
    akhir = tengah - 1;
} else {
    awal = tengah + 1;
}
}
if(!ditemukan) {
    System.out.println("Data dosen dengan usia " + usia + " tidak ditemukan!");
}
}
}
}

```

## MahasiswaDemo

```

import java.util.Scanner;

public class DosenDemo17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen17 data = new DataDosen17(100); // Inisialisasi dengan kapasitas maksimal 100
        int menu;

        do {
            System.out.println("\n=====");
            System.out.println("SISTEM PENGELOLAAN DATA DOSEN");
            System.out.println("=====");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampil Data Dosen");
            System.out.println("3. Cari Data Dosen Menggunakan Nama - Sequential Search");
            System.out.println("4. Cari Data Dosen Menggunakan Usia - Binary Search");
            System.out.println("0. Keluar");
            System.out.println("=====");
            System.out.print("Pilih Menu (0-5): ");
            menu = sc.nextInt();
            sc.nextLine();

            switch(menu) {
                case 1:
                    System.out.println("\nTambah Data Dosen");
                    System.out.println("=====");
                    System.out.print("Masukkan jumlah dosen yang akan ditambahkan: ");
                    int jumlah = sc.nextInt();
                    sc.nextLine();

                    for(int i = 0; i < jumlah; i++) {
                        System.out.println("\nMasukkan data dosen ke-" + (i+1));
                        System.out.print("Masukkan nama: ");
                        String nama = sc.nextLine();
                        System.out.print("Masukkan usia: ");
                        int usia = sc.nextInt();
                        sc.nextLine();
                        System.out.print("Masukkan NIP: ");
                        String nip = sc.nextLine();
                        System.out.print("Masukkan jenis kelamin (L/P): ");
                        String jenisKelamin = sc.nextLine();

                        Dosen17 d = new Dosen17(nama, usia, nip, jenisKelamin);
                        data.tambah(d);
                    }
                    break;

                case 2:
                    System.out.println("\nData Dosen:");
                    System.out.println("=====");
                    data.tampil();
                    break;

                case 3:
                    System.out.println("\nPencarian Data Sequential Search");
                    System.out.println("=====");
                    System.out.print("Masukkan nama dosen yang dicari: ");
                    String cariNama = sc.nextLine();
                    data.pencarianDataSequential(cariNama);
                    break;
            }
        } while (menu != 0);
    }
}

```

```

case 4:
    System.out.println("\nPencarian Data Binary Search");
    System.out.println("=====");
    System.out.print("Masukkan usia dosen yang dicari: ");
    int cariUsia = sc.nextInt();
    data.pencarianDataBinary(cariUsia);
    break;

case 0:
    System.out.println("\nTerima kasih telah menggunakan program ini!");
    break;

default:
    System.out.println("\nMenu tidak valid!");
}
} while(menu != 0);
}
}

```

#### - Hasil Kode Program

```

=====
SISTEM PENGELOLAAN DATA DOSEN
=====
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Cari Data Dosen Menggunakan Nama - Sequential Search
4. Cari Data Dosen Menggunakan Usia - Binary Search
0. Keluar
=====
Pilih Menu (0-5): 1

Tambah Data Dosen
=====
Masukkan jumlah dosen yang akan ditambahkan: 4

Masukkan data dosen ke-1
Masukkan nama: Budi
Masukkan usia: 56
Masukkan NIP: 111
Masukkan jenis kelamin (L/P): L

Masukkan data dosen ke-2
Masukkan nama: Agus
Masukkan usia: 34
Masukkan NIP: 222
Masukkan jenis kelamin (L/P): L

Masukkan data dosen ke-3
Masukkan nama: Putri
Masukkan usia: 34
Masukkan NIP: 333
Masukkan jenis kelamin (L/P): P

Masukkan data dosen ke-4
Masukkan nama: Mustafid
Masukkan usia: 32
Masukkan NIP: 444
Masukkan jenis kelamin (L/P): L

```

```

=====
SISTEM PENGELOLAAN DATA DOSEN
=====
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Cari Data Dosen Menggunakan Nama - Sequential Search
4. Cari Data Dosen Menggunakan Usia - Binary Search
0. Keluar
=====
Pilih Menu (0-5): 2

Data Dosen:
=====

Data Dosen ke-1
Nama: Budi
Usia: 56
NIP: 111
Jenis Kelamin: L
-----

Data Dosen ke-2
Nama: Agus
Usia: 34
NIP: 222
Jenis Kelamin: L
-----

Data Dosen ke-3
Nama: Putri
Usia: 34
NIP: 333
Jenis Kelamin: P
-----

Data Dosen ke-4
Nama: Mustafid
Usia: 32
NIP: 444
Jenis Kelamin: L
-----

```

```

=====
SISTEM PENGELOLAAN DATA DOSEN
=====
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Cari Data Dosen Menggunakan Nama - Sequential Search
4. Cari Data Dosen Menggunakan Usia - Binary Search
0. Keluar
=====
Pilih Menu (0-5): 3

Pencarian Data Sequential Search
=====
Masukkan nama dosen yang dicari: Mustafid

Hasil Pencarian untuk nama: Mustafid
=====
Data ditemukan pada indeks berikut:

Data ke-1 pada indeks 3
Nama: Mustafid
Usia: 32
NIP: 444
Jenis Kelamin: L

```

```

=====
SISTEM PENGELOLAAN DATA DOSEN
=====
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Cari Data Dosen Menggunakan Nama - Sequential Search
4. Cari Data Dosen Menggunakan Usia - Binary Search
0. Keluar
=====
Pilih Menu (0-5): 4

Pencarian Data Binary Search
=====
Masukkan usia dosen yang dicari: 34

Hasil Pencarian untuk usia: 34
=====
Data ditemukan pada indeks berikut:

Data ke-1 pada indeks 1
Nama: Agus
Usia: 34
NIP: 222
Jenis Kelamin: L

Data ke-2 pada indeks 2
Nama: Putri
Usia: 34
NIP: 333
Jenis Kelamin: P

[PERINGATAN] Ditemukan 2 data dengan usia yang sama!
Silakan gunakan informasi tambahan (seperti nama atau NIP) untuk identifikasi lebih spesifik.

=====
SISTEM PENGELOLAAN DATA DOSEN
=====
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Cari Data Dosen Menggunakan Nama - Sequential Search
4. Cari Data Dosen Menggunakan Usia - Binary Search
0. Keluar
=====
Pilih Menu (0-5): 0

Terima kasih telah menggunakan program ini!
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD>

```