

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN STRUKTUR  
DASAR**

**Jobsheet 7**



**Adi Luhung**

**244107020088**

**Kelas 1E**

**Program Studi Teknik Informatika**

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

**2025**

## 1. Pratikum

### 1.1 Percobaan 1 – Sequential Search

1. Membuat kode program untuk mengelola data mahasiswa yang terdiri dari menambah data, menampilkan seluruh data, mencari data, dan menampilkan letak data.

```
package Jobsheet7;
import java.util.Scanner;
public class MahasiswaDemo01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int jumlahMahasiswa = 5;
        MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01(jumlahMahasiswa);

        for (int i = 0; i < jumlahMahasiswa; 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       : ");
            double ipk = sc.nextDouble();
            System.out.println("-----");
            sc.nextLine();

            Mahasiswa01 m = new Mahasiswa01(nim, nama, kelas, ipk);
            list.tambah(m);
        }

        list.tampil();
        System.out.println("-----");
        System.out.println("Percarian Data");
        System.out.println("-----");
        System.out.println("Masukkan IPK mahasiswa yang akan dicari: ");
        System.out.print("IPK: ");
        double cari = sc.nextDouble();

        System.out.println("Menggunakan sequential searching");
        double posisi = list.sequentialSearch(cari);
        int pss = (int) posisi;
        list.tampilPosisi(cari, pss);
        list.tampilDataSearch(cari, pss);
        sc.close();
    }
}
```

```

package Jobsheet7;
public class MahasiswaBerprestasi01 {
    Mahasiswa01[] listMhs;
    int idx;
    MahasiswaBerprestasi01(int jumlahMahasiswa) {
        listMhs = new Mahasiswa01[jumlahMahasiswa];
        idx = 0;
    }
    void tambah(Mahasiswa01 mhs) {
        if (idx < listMhs.length) {
            listMhs[idx] = mhs;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }
    void tampil() {
        for (Mahasiswa01 mhs : listMhs) {
            if (mhs != null) {
                mhs.tampilInformasi();
                System.out.println("-----");
            }
        }
    }
    int sequentialSearch(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");
        }
    }
}

```

```

package Jobsheet7;
public class Mahasiswa01 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa01() {
    }
    Mahasiswa01(String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }
    void tampilInformasi() {
        System.out.println("Nama      : " + nama);
        System.out.println("NIM      : " + nim);
        System.out.println("Kelas   : " + kelas);
        System.out.println("IPK      : " + ipk);
    }
}

```

## 2. 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

Percarian Data

Masukkan IPK mahasiswa yang akan 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

## 1.2 Pertanyaan

1. Fungsi dari *method* `tampilPosisi()` adalah untuk menampilkan indeks dari array `listMhs` hasil dari pencarian yang dilakukan *method* `sequentialSearch()`. Sedangkan *method* `tampilDataSearch()` berfungsi untuk menampilkan detail isi dari indeks yang dihasilkan oleh *method* `sequentialSearch()`.
2. Fungsi dari *break* pada *method* `sequentialSearch()` adalah untuk menghentikan perulangan yang terjadi setelah nilai dari variabel `cari` sama dengan nilai yang terdapat pada `listMhs[j].ipk`.

## 2.1 Percobaan 2 – Binary - Search

1. Memodifikasi kode program dengan menambahkan *method* `findBinarySearch()`

```
package Jobsheet7;
public class Mahasiswa01 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa01() {
    }
    Mahasiswa01(String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }
    void tampilInformasi() {
        System.out.println("Nama      : " + nama);
        System.out.println("NIM      : " + nim);
        System.out.println("Kelas   : " + kelas);
        System.out.println("IPK      : " + ipk);
    }
}
```

```

package Jobsheet7;
public class MahasiswaBerprestasi01 {
    Mahasiswa01[] listMhs;
    int idx;
    MahasiswaBerprestasi01(int jumlahMahasiswa) {
        listMhs = new Mahasiswa01[jumlahMahasiswa];
        idx = 0;
    }
    void tambah(Mahasiswa01 mhs) {
        if (idx < listMhs.length) {
            listMhs[idx] = mhs;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }
    void tampil() {
        for (Mahasiswa01 mhs : listMhs) {
            if (mhs != null) {
                mhs.tampilInformasi();
                System.out.println("-----");
            }
        }
    }
    int sequentialSearch(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;
    }
}

```

```

package Jobsheet7;
import java.util.Scanner;
public class MahasiswaDemo01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int jumlahMahasiswa = 5;
        MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01(jumlahMahasiswa);

        for (int i = 0; i < jumlahMahasiswa; 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      : ");
            double ipk = sc.nextDouble();
            System.out.println("-----");
            sc.nextLine();

            Mahasiswa01 m = new Mahasiswa01(nim, nama, kelas, ipk);
            list.tambah(m);
        }
        list.tampil();
        // System.out.println("-----");
        // System.out.println("Pencarian Data");
        // System.out.println("-----");
        // System.out.println("Masukkan IPK mahasiswa yang akan dicari: ");
        // System.out.print("IPK: ");
        // double cari = sc.nextDouble();

        // System.out.println("Menggunakan sequential searching");
        // double posisi = list.sequentialSearch(cari);
        // int pss = (int) posisi;
        // list.tampilPosisi(cari, pss);
        // list.tampilDataSearch(cari, pss);

        System.out.println("-----");
        // System.out.println("Pencarian Data");
        System.out.println("-----");
        // System.out.println("Masukkan IPK mahasiswa yang akan 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, jumlahMahasiswa-1);
        int pss2 = (int) posisi2;
        list.tampilPosisi(cari, pss2);
        list.tampilDataSearch(cari, pss2);
        sc.close();
    }
}

```

## 2. Hasil dari kode program

```
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 akan 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
```



## 2.2 Pertanyaan

1. Proses *divide* terjadi pada method `findBinarySearch()` tepatnya pada kode program bagian:

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

2. Proses *qonquer* terjadi pada method `findBinarySearch()` tepatnya pada kode program bagian:

```
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 (sesuai dengan logika yang dipakai pada kode program), maka kemungkinan data IPK yang ingin dicari tidak ditemukan. Seperti contoh dibawah ini:

```
Masukkan data mahasiswa ke-1:  
NIM    : 111  
Nama   : adi  
Kelas : 2  
IPK    : 3,7  
-----  
Masukkan data mahasiswa ke-2:  
NIM    : 222  
Nama   : ila  
Kelas : 2  
IPK    : 3,1  
-----  
Masukkan data mahasiswa ke-3:  
NIM    : 333  
Nama   : lia  
Kelas : 2  
IPK    : 3,5  
-----  
Masukkan data mahasiswa ke-4:  
NIM    : 333  
Nama   : susi  
Kelas : 2  
IPK    : 3,2  
-----  
Masukkan data mahasiswa ke-5:  
NIM    : 555  
Nama   : anita  
Kelas : 2  
IPK    : 3,3  
-----
```

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

```

-----
Percarian Data
-----
Masukkan IPK mahasiswa yang akan dicari:
IPK: 3,7
-----
Menggunakan binary search
-----
Data 3.7 tidak ditemukan
Data mahasiswa dengan IPK 3.7 tidak ditemukan

```

4. Mencoba memasukkan data secara *descending* dan memodifikasi program.
  - a. Memasukan data IPK secara *descending*.

<pre> Masukkan data mahasiswa ke-1: NIM    : 111 Nama   : adi Kelas : 2 IPK    : 3,7 ----- Masukkan data mahasiswa ke-2: NIM    : 222 Nama   : ila Kelas : 2 IPK    : 3,5 ----- 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    : 3,2 ----- Masukkan data mahasiswa ke-5: NIM    : 555 Nama   : anita Kelas : 2 IPK    : 3,1 ----- </pre>	<pre> Nama   : adi NIM    : 111 Kelas : 2 IPK    : 3.7 ----- Nama   : ila NIM    : 222 Kelas : 2 IPK    : 3.5 ----- Nama   : lia NIM    : 333 Kelas : 2 IPK    : 3.3 ----- Nama   : susi NIM    : 444 Kelas : 2 IPK    : 3.2 ----- Nama   : anita NIM    : 555 Kelas : 2 IPK    : 3.1 ----- </pre>
--	--

```

-----
Percarian Data
-----
Masukkan IPK mahasiswa yang akan dicari:
IPK: 3,7
-----
Menggunakan binary search
-----
Data 3.7 tidak ditemukan
Data mahasiswa dengan IPK 3.7 tidak ditemukan

```

Jika memasukkan data IPK secara *descending*, maka IPK tidak akan ditemukan. Sama seperti saat data dimasukkan secara acak. Hal ini dikarenakan terdapat perbandingan antara data didalam kode program yang sudah diatur untuk data yang terurut secara *ascending*.

- b. Memodifikasi kode program pada method `findBinarySearch()` agar dapat menampilkan hasil yang sesuai saat memasukkan data secara *descending*.

```
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;
}
```

5. Memodifikasi kode program agar jumlah data mahasiswa yang ingin dimasukkan dapat diatur sesuai input dari keyboard.

```
package Jobsheet7;
import java.util.Scanner;
public class MahasiswaDemo01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan jumlah data mahasiswa: ");
        int jumlahMahasiswa = sc.nextInt();
        sc.nextLine();
        MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01(jumlahMahasiswa);
    }
}
```

Hasil kode program

```
Masukkan jumlah data mahasiswa: 3
Masukkan data mahasiswa ke-1:
NIM      : 111
Nama     : adi
Kelas   : 2
IPK      : 3,8
-----
Masukkan data mahasiswa ke-2:
NIM      : 222
Nama     : lia
Kelas   : 2
IPK      : 3,4
-----
Masukkan data mahasiswa ke-3:
NIM      : 333
Nama     : ila
Kelas   : 2
IPK      : 3,2
-----
Nama      : adi
NIM       : 111
Kelas    : 2
IPK       : 3.8
-----
Nama      : lia
NIM       : 222
Kelas    : 2
IPK       : 3.4
-----
Nama      : ila
NIM       : 333
Kelas    : 2
IPK       : 3.2
-----
Pencarian Data
-----
Masukkan IPK mahasiswa yang akan dicari:
IPK: 3,8
-----
Menggunakan binary search
-----
Data mahasiswa dengan IPK : 3.8 ditemukan pada indeks 0
NIM      : 111
Nama     : adi
Kelas   : 2
IPK      : 3.8
```

## 2. Latihan Praktikum

- a. Memodifikasi kode program tentang data dosen pada percobaan sebelumnya dengan menambahkan method `PencarianDataSequential01()` :

```
int pencarianDataSequential01(int cari) {
    for (int j = 0; j < listDosen.length; j++) {
        if (listDosen[j].usia == cari) {
            return j;
        }
    }
    return -1;
}
```

- b. Memodifikasi kode program tentang data dosen pada percobaan sebelumnya dengan menambahkan method `PencarianDataBinary01()` :

```
int pencarianDataBinary01(int cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (left + right) / 2;
        if (cari == listDosen[mid].usia) {
            return mid;
        } else if (listDosen[mid].usia < cari) {
            return pencarianDataBinary01(cari, mid + 1, right);
        } else {
            return pencarianDataBinary01(cari, left, mid - 1);
        }
    }
    return -1;
}
```

- c. Menambahkan aturan untuk mendeteksi hasil pencarian lebih dari 1 dalam bentuk peringatan

```

package Jobsheet7;
import java.util.Scanner;
public class DosenDemo01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen01 list = new DataDosen01(3);
        for (int i = 0; i < list.listDosen.length ; i++) {
            System.out.println("Masukkan data dosen ke-" + (i + 1) + ":");
            System.out.print("Kode          : ");
            String kode = sc.nextLine();
            System.out.print("Nama          : ");
            String nama = sc.nextLine();
            System.out.print("Jenis Kelamin : ");
            String jk = sc.nextLine();
            System.out.print("Usia          : ");
            int usia = sc.nextInt();
            sc.nextLine();
            System.out.println("-----");

            Boolean jenisKelamin = false;
            if (jk.equalsIgnoreCase("pria")) {
                jenisKelamin = true;
            }

            Dosen01 dsn = new Dosen01(kode, nama, jenisKelamin, usia);
            list.tambah(dsn);
        }
        System.out.println("\nData dosen sebelum sorting: ");
        list.tampil();

        System.out.println("\nData dosen setelah sorting berdasarkan Usia (ASC): ");
        list.sortingASC();
        list.tampil();

        System.out.println("-----");
        System.out.println("Pencarian Data");
        System.out.println("-----");
        System.out.println("Masukkan usia dosen yang akan dicari: ");
        System.out.print("Usia: ");
        int cari = sc.nextInt();
        System.out.println("-----");
        System.out.println("Menggunakan sequential search");
        System.out.println("-----");
        int posisi = list.pencarianDataSequential01(cari);
        list.tampilDataSearch(cari, posisi);
        System.out.println(" ");

        System.out.println("-----");
        System.out.println("Pencarian Data");
        System.out.println("-----");
        System.out.println("Masukkan usia dosen yang akan dicari: ");
        System.out.print("Usia: ");
        cari = sc.nextInt();
        System.out.println("-----");
        System.out.println("Menggunakan binary search");
        System.out.println("-----");
        posisi = list.pencarianDataBinary01(cari, 0, list.listDosen.length - 1);
        list.tampilDataSearch(cari, posisi);
        sc.close();
    }
}

```

```

package Jobsheet7;
public class DataDosen01 {
    Dosen01[] listDosen;
    int idx;
    DataDosen01 (int n) {
        listDosen = new Dosen01 [n];
        idx = 0;
    }
    void tambah(Dosen01 dsn) {
        if (idx < listDosen.length) {
            listDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }
    void tampil() {
        for (Dosen01 dsn : listDosen) {
            if (dsn != null) {
                dsn.tampil();
                System.out.println("-----");
            }
        }
    }
    void sortingASC() {
        for (int i = 0; i < listDosen.length - 1; i++) {
            for (int j = 1; j < listDosen.length - i; j++) {
                if (listDosen[j].usia < listDosen[j - 1].usia) {
                    Dosen01 tmp = listDosen[j];
                    listDosen[j] = listDosen[j - 1];
                    listDosen[j - 1] = tmp;
                }
            }
        }
    }
    int hasil = 0;
    int pencarianDataSequential01(int cari) {
        hasil = 0;
        int posisiPertama = -1;

        for (int j = 0; j < listDosen.length; j++) {
            if (listDosen[j] != null && listDosen[j].usia == cari) {
                if (posisiPertama == -1) {
                    posisiPertama = j;
                }
                hasil++;
            }
        }
        return posisiPertama;
    }
}

```

```

int pencarianDataBinary01(int cari, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;

        if (listDosen[mid].usia == cari) {
            hasil = 1;

            int i = mid - 1;
            while (i >= 0 && listDosen[i].usia == cari) {
                hasil++;
                i--;
            }

            int j = mid + 1;
            while (j < listDosen.length && listDosen[j].usia == cari) {
                hasil++;
                j++;
            }

            return mid;
        } else if (listDosen[mid].usia < cari) {
            return pencarianDataBinary01(cari, mid + 1, right);
        } else {
            return pencarianDataBinary01(cari, left, mid - 1);
        }
    }
    hasil = 0;
    return -1;
}

void tampilDataSearch(int usia, int pos) {
    if (hasil > 1) {
        System.out.println("Peringatan! Ditemukan lebih dari satu dosen dengan usia "
+ usia + ".");
    }

    if (hasil > 0) {
        System.out.println("Menampilkan semua dosen dengan usia " + usia + ":");
        for (int i = 0; i < listDosen.length; i++) {
            if (listDosen[i] != null && listDosen[i].usia == usia) {
                System.out.println("-----");
                System.out.println("Kode\t\t : " + listDosen[i].kode);
                System.out.println("Nama\t\t : " + listDosen[i].nama);
                System.out.println("Jenis Kelamin : " + listDosen[i].kelamin());
                System.out.println("Usia\t\t : " + listDosen[i].usia);
            }
        }
    } else {
        System.out.println("Data dosen dengan usia " + usia + " tidak ditemukan");
    }
}
}

```

```

package Jobsheet7;
public class Dosen01 {
    String kode;
    String nama;
    Boolean jenisKelamin;
    int usia;

    Dosen01 () {
    }
    Dosen01 (String kd, String name, Boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }
    String kelamin() {
        if (jenisKelamin == true) {
            return "Pria";
        } else {
            return "Wanita";
        }
    }
    void tampil() {
        kelamin();
        System.out.println("Kode           : " + kode);
        System.out.println("Nama           : " + nama);
        System.out.println("Jenis Kelamin  : " + kelamin());
        System.out.println("Usia          : " + usia);
    }
}

```

### Hasil kode program

Masukkan data dosen ke-1:

```

Kode       : 111
Nama       : Mawar
Jenis Kelamin : Wanita
Usia       : 25
-----

```

Masukkan data dosen ke-2:

```

Kode       : 222
Nama       : Budi
Jenis Kelamin : Pria
Usia       : 47
-----

```

Masukkan data dosen ke-3:

```

Kode       : 333
Nama       : Bagus
Jenis Kelamin : Pria
Usia       : 25
-----

```

Data dosen sebelum sorting:

```

Kode       : 111
Nama       : Mawar
Jenis Kelamin : Wanita
Usia       : 25
-----

```

```

Kode       : 222
Nama       : Budi
Jenis Kelamin : Pria
Usia       : 47
-----

```

```

Kode       : 333
Nama       : Bagus
Jenis Kelamin : Pria
Usia       : 25
-----

```



Data dosen setelah sorting berdasarkan Usia (ASC):

Kode : 111  
Nama : Mawar  
Jenis Kelamin : Wanita  
Usia : 25

-----  
Kode : 333  
Nama : Bagus  
Jenis Kelamin : Pria  
Usia : 25

-----  
Kode : 222  
Nama : Budi  
Jenis Kelamin : Pria  
Usia : 47  
-----

-----  
Percarian Data

Masukkan usia dosen yang akan dicari:

Usia: 25

-----  
Menggunakan sequential search

-----  
Peringatan! Ditemukan lebih dari satu dosen dengan usia 25.  
Menampilkan semua dosen dengan usia 25:

-----  
Kode : 111  
Nama : Mawar  
Jenis Kelamin : Wanita  
Usia : 25

-----  
Kode : 333  
Nama : Bagus  
Jenis Kelamin : Pria  
Usia : 25

-----  
Percarian Data

Masukkan usia dosen yang akan dicari:

Usia: 47

-----  
Menggunakan binary search

-----  
Menampilkan semua dosen dengan usia 47:

-----  
Kode : 222  
Nama : Budi  
Jenis Kelamin : Pria  
Usia : 47

- d. Push semua hasil kode program ke github

```
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> ls

Directory: C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD

Mode                LastWriteTime         Length Name
----                -
d-----          3/13/2025   7:50 AM                BruteForceDivideQonquer
d-----          2/13/2025   9:52 PM                Jobsheet1
d-----          2/20/2025   8:06 PM                Jobsheet2
d-----          2/27/2025   9:45 AM                Jobsheet3
d-----          4/8/2025    2:14 PM                Jobsheet7
d-----          3/27/2025  10:26 PM                Praktikum5

PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git add .
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git commit -m "Jobsheet 7"
[main 6818dbd] Jobsheet 7
6 files changed, 341 insertions(+)
create mode 100644 Jobsheet7/DataDosen01.java
create mode 100644 Jobsheet7/Dosen01.java
create mode 100644 Jobsheet7/DosenDemo01.java
create mode 100644 Jobsheet7/Mahasiswa01.java
create mode 100644 Jobsheet7/MahasiswaBerprestasi01.java
create mode 100644 Jobsheet7/MahasiswaDemo01.java
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 3.49 KiB | 3.49 MiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote:   https://github.com/Kumaaan/Praktikum_ASD.git
To https://github.com/Kumaaan/Praktikum-ASD.git
   d4cd094..6818dbd  main -> main
```