

ALGORITMA DAN STRUKTUR DATA

“Laporan hasil Praktikum pada Jobsheet 7 “SEARCHING” ”

Oleh:

Hafiz Rizqi Hernanda

NIM (244107020154)



Jurusan Teknologi informasi

Teknik Informatika

Politeknik Negeri Malang

7.2. Searching / Pencarian Menggunakan Agoritma Sequential Search

Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa.

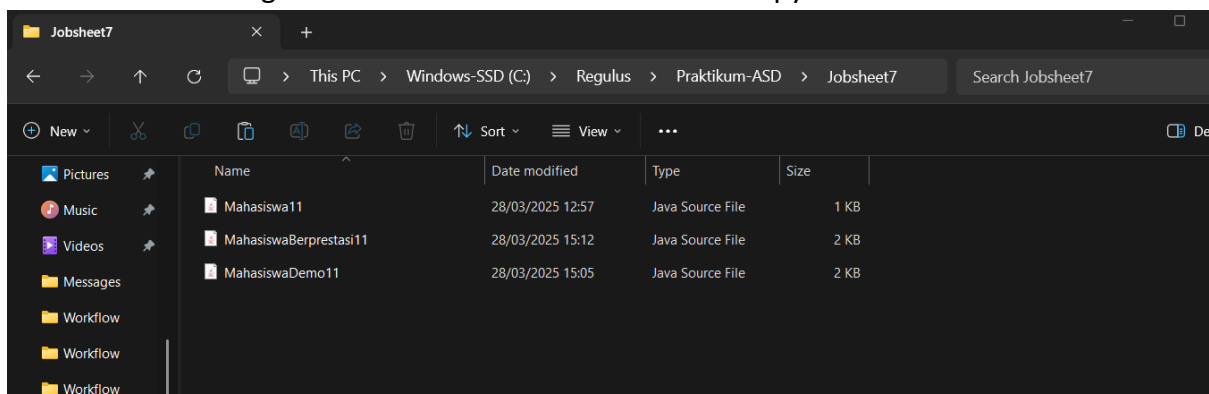
Mahasiswa
nim: String nama: String kelas: String ipk: double
Mahasiswa() Mahasiswa(nm: String, name: String, kls: String, ip: double) tampilInformasi(): void

Berdasarkan class diagram di atas, akan dibuat class Mahasiswa yang berfungsi untuk membuat objek mahasiswa yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga fungsi tampilInformasi() untuk menampilkan semua attribute yang ada.

MahasiswaBerprestasi
listMhs: Mahasiswa[5] idx: int
tambah(mhs: Mahasiswa): void tampil(): void sequentialSearch(double cari): int tampilPosisi(double x,int pos): void tampilDataSearch(double x,int pos):void

7.2.1. Langkah-langkah Percobaan Sequential Search

1. Pada pertemuan Jobsheet 7 ini akan menggunakan class Mahasiswa, MahasiswaBerprestasi, dan MahasiswaDemo pada pertemuan Jobsheet 6 sebelumnya
2. Buat folder baru bernama Jobsheet7 di dalam repository Praktikum ASD, kemudian buka ketiga class dari Jobsheet 6 tersebut dan copy ke folder Jobsheet 7



3. Tambahkan method sequentialSearching bertipe integer dengan parameter cari bertipe double pada class MahasiswaBerprestasi. Kemudian Deklarasikan isi method sequentialSearching dengan algoritma pencarian data menggunakan teknik sequential searching.

```

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;
}

```

4. Buatlah method tampilPosisi bertipe void dan Deklarasikan isi dari method tampilPosisi pada class MahasiswaBerprestasi.

```

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

```

5. Pada class MahasiswaBerprestasi, buatlah method tampilDataSearch bertipe void dan Deklarasikan isi dari method tampilDataSearch .

```

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");
    }
}

```

6. Pada class MahasiswaDemo , tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.

```

J MahasiswaDemo11.java > MahasiswaDemo11 > main(String[])
1  import java.util.Scanner;
2  public class MahasiswaDemo11 {
    Run | Debug
3      public static void main(String[] args) {
4          Scanner input = new Scanner(System.in);
5          MahasiswaBerprestasi11 list = new MahasiswaBerprestasi11();
6          int jmlMhs = 5;
7
8          for (int i = 0; i < jmlMhs; i++) {
9              System.out.println("Masukkan data mahasiswa ke-" + (i + 1) );
10             System.out.print(s:"NIM: ");
11             String nim = input.nextLine();
12             System.out.print(s:"Nama: ");
13             String nama = input.nextLine();
14             System.out.print(s:"Kelas: ");
15             String kelas = input.nextLine();
16             System.out.print(s:"IPK: ");
17             double ipk = input.nextDouble();
18             input.nextLine();
19             System.out.println(x:"-----");
20
21             Mahasiswa11 mhs = new Mahasiswa11(nim, nama, kelas, ipk);
22             list.tambah(mhs);
23         }
24
25         list.tampil();
26         //melakukan pencarian data sequential
27         System.out.println(x:"-----");
28         System.out.println(x:"Pencarian data");
29         System.out.println(x:"-----");
30         System.out.println(x:"masukkan ipk mahasiswa yang dicari: ");
31         System.out.print(s:"IPK: ");
32         double cari = input.nextDouble();
33
34         System.out.println(x:"menggunakan sequential searching");
35         double posisi = list.sequentialSearching(cari);
36         int pss = (int) posisi;
37         list.tampilPosisi(cari, pss);
38         list.tampilDataSearch(cari, pss);
39     }
40 }
41

```

7. Jalankan dan amati hasilnya.

7.2.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini

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

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 index ke-3
nim      : 444
nama     : lia
kelas    : 2
ipk      : 3.5
PS C:\Regulus\Praktikum-ASD\Jobsheet7>

```

7.2.3. Pertanyaan

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

Perbedaan nya ada pada tujuan dari method itu untuk menampilkan sebuah informasi dan data yang diberikan, pada method tampilPosisi berfungsi menampilkan informasi berupa posisi dari data mahasiswa yang memiliki IPK (x),method ini menampilkan “apakah data ini ditemukan dan di indeks ke berapa” sedangkan methodtampilDataSearch berfungsi menampilkan detail data mahasiswa yang memiliki IPK (x), jika ditemukan, method ini menampilkan (nama, nim, kelas, ipk).

2. Jelaskan fungsi break pada kode program dibawah ini!

Untuk menghentikan perulangan pada for secara paksa, jika tidak ada fungsi break perulangan pada for akan terus berjalan sampai akhir array, walaupun data sudah ditemukan.

7.3. Searching / Pencarian Menggunakan Binary Search

7.3.1. Langkah-langkah Percobaan Binary Search

1. Pada percobaan 6.2.1 (sequential search) tambahkan method findBinarySearch bertipe integer pada class MahasiswaBerprestasi. Kemudian Deklarasikan isi method findBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.

```
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;
}
```

2. Panggil method findBinarySearch terdapat pada class MahasiswaBerprestasi di kelas MahasiswaDemo. Kemudian panggil method tampilPosisi dan tampilDataSearch

```
list.tampil();
//melakukan pencarian data Binary
System.out.println(x:"-----");
System.out.println(x:"Pencarian data");
System.out.println(x:"-----");
System.out.println(x:"masukkan ipk mahasiswa yang dicari: ");
System.out.print(s:"IPK: ");
double cari = input.nextDouble();

System.out.println(x:"menggunakan binary search");
double posisi2 = list.findBinarySearch(cari, left:0, jmlMhs-1);
int pss2 = (int) posisi2;
list.tampilPosisi(cari, pss2);
list.tampilDataSearch(cari, pss2);
```

3. Jalankan dan amati hasilnya (inputkan data IPK secara terurut -ASC seperti verifikasi hasil percobaan dibawah ini).

7.3.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini.

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

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 index ke-4
nim      : 555
nama     : anita
kelas    : 2
ipk      : 3.7
PS C:\Regulus\Praktikum-ASD\Jobsheet7>

```

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!

```

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 bisa di jalankan, karena Binary search hanya bekerja dengan array yang sudah terurut(ascending atau descending)

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

```
Kelas: 2
IPK: 3,2
-----
Nama: adi
NIM: 111
Kelas: 2
IPK: 3.8
-----
Nama: ila
NIM: 22
Kelas: 2
IPK: 3.7
-----
Nama: lia
NIM: 333
Kelas: 2
IPK: 3.5
-----
Nama: susi
NIM: 444
Kelas: 2
IPK: 3.4
-----
Nama: anita
NIM: 555
Kelas: 2
IPK: 3.2
-----
-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,2
menggunakan binary search
data 3.2 tidak ditemukan
Data mahasiswa dengan IPK 3.2 tidak ditemukan
PS C:\Regulus\Praktikum-ASD\Jobsheet7> █
```

Tidak sesuai, maka pada kode `else if(listMhs[mid].ipk > cari)` di ubah `else if(listMhs[mid].ipk < cari)`

```
Nama: adi
NIM: 111
Kelas: 2
IPK: 3.8
-----
Nama: ila
NIM: 222
Kelas: 2
IPK: 3.7
-----
Nama: lia
NIM: 333
Kelas: 2
IPK: 3.5
-----
Nama: susi
NIM: 444
Kelas: 2
IPK: 3.4
-----
Nama: anita
NIM: 555
Kelas: 2
IPK: 3.2
-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,2
menggunakan binary search
data mahasiswa dengan IPK: 3.2 ditemukan pada index ke-4
nim      : 555
nama     : anita
kelas    : 2
ipk      : 3.2
PS C:\Regulus\Praktikum-ASD\Jobsheet7>
```

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

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    MahasiswaBerprestasi11 list = new MahasiswaBerprestasi11();
    System.out.print(s:"Masukkan Jumlah Mahasiswa: ");
    int jmlMhs = input.nextInt();
    input.nextLine();

    for (int i = 0; i < jmlMhs; i++) {
        System.out.println("Masukkan data mahasiswa ke-" + (i + 1) );
        System.out.print(s:"NIM: ");
        String nim = input.nextLine();
        System.out.print(s:"Nama: ");
        String nama = input.nextLine();
        System.out.print(s:"Kelas: ");
        String kelas = input.nextLine();
        System.out.print(s:"IPK: ");
        double ipk = input.nextDouble();
        System.out.println(x:"-----");
        input.nextLine();
    }
}

```

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!

```

1 public class DataDosen11 {
2     void insertionSort() {
3
4     }
5
6     int sequentialSearching(String cari) {
7         int count = 0;
8         int posisi = -1;
9         for (int j = 0; j < idx; j++) {
10             if (data11[j].nama.equalsIgnoreCase(cari)) {
11                 if (count == 0) posisi = j; // simpan posisi pertama ditemukan
12                 count++;
13             }
14         }
15
16         if (count > 1) {
17             System.out.println(x:" Peringatan: Ditemukan lebih dari satu dosen dengan nama yang sama!");
18         }
19
20         return posisi;
21     }
22
23     int findBinarySearch(int cari, int left, int right) {
24         if (right >= left) {
25             int mid;
26             mid = (left + right) / 2;
27             if (cari == data11[mid].usia) {
28                 return mid;
29             } else if (data11[mid].usia > cari) {
30                 return findBinarySearch(cari, left, mid - 1);
31             } else {
32                 return findBinarySearch(cari, mid + 1, right);
33             }
34         }
35         return -1;
36     }
37 }

```

```

int binarySearchDenganPeringatan(int usia) {
    int count = 0;
    int pos = findBinarySearch(usia, left:0, idx - 1); // pakai binary search dulu

    if (pos != -1) {
        count = 1;

        // Cek ke kiri
        int left = pos - 1;
        while (left >= 0 && data11[left].usia == usia) {
            count++;
            left--;
        }

        // Cek ke kanan
        int right = pos + 1;
        while (right < idx && data11[right].usia == usia) {
            count++;
            right++;
        }

        if (count > 1) {
            System.out.println(x:"Peringatan: Ditemukan lebih dari satu dosen dengan usia yang sama!");
        }
    }

    return pos; // tetap kembalikan posisi awal yang ditemukan
}

void tampilPosisi(String x, int pos) {
    if (pos != -1) {
        System.out.println("Data dosen dengan nama: " + x + " ditemukan pada index ke-" + pos);
    } else {
        System.out.println("Data " + x + " tidak ditemukan");
    }
}

void tampilDataSearch(String x, int pos) {
    if (pos != -1) {
        System.out.println("kode\t: " + data11[pos].kode);
        System.out.println("nama\t: " + data11[pos].nama);
        System.out.println("jenis kelamin\t: " + (data11[pos].jenisKelamin ? "Laki-laki" : "Perempuan"));
        System.out.println("usia\t: " + data11[pos].usia);
    } else {
        System.out.println("Data dosen dengan nama " + x + " tidak ditemukan");
    }
}
}

```



```

J DosenMain11.java > DosenMain11 > main(String[])
2 public class DosenMain11 {
3     public static void main(String[] args) {
4         System.out.println(x:"4. Tampilkan Data Dosen ");
12        System.out.println(x:"3. Sorting ASC (Bubble Sort)");
13        System.out.println(x:"4. Sorting DSC (Selection Sort)");
14        System.out.println(x:"5. Sorting DSC (Insertion Sort)");
15        System.out.println(x:"6. Sequential Search dengan Nama Dosen");
16        System.out.println(x:"7. Binary Search dengan Usia Dosen (ASC)");
17        System.out.println(x:"8. Keluar");
18        System.out.print(s:"Pilih menu: ");
19        pilihan = sc.nextInt();
20        sc.nextLine();
21
22        switch (pilihan) {
23            case 1:
24                System.out.print(s:"Masukkan kode: ");
25                String kode = sc.nextLine();
26                System.out.print(s:"Masukkan nama: ");
27                String nama = sc.nextLine();
28                System.out.print(s:"Masukkan jenis kelamin (true untuk laki-laki, false untuk perempuan): ");
29                boolean jk = sc.nextBoolean();
30                System.out.print(s:"Masukkan usia: ");
31                int usia = sc.nextInt();
32                dsn.tambah(new Dosen11(kode, nama, jk, usia));
33                break;
34            case 2:
35                dsn.tampil();
36                break;
37            case 3:
38                dsn.SortingASC();
39                dsn.tampil();
40                break;
41            case 4:
42                dsn.SortingDSC();
43                dsn.tampil();
44                break;
45            case 5:
46                dsn.insertionSort();
47                dsn.tampil();
48                break;
49            case 6:
50                System.out.print(s:"Masukkan nama dosen yang dicari: ");
51                String cari = sc.nextLine();
52                int pos = dsn.sequentialSearching(cari);
53                dsn.tampilPosisi(cari, pos);
54                dsn.tampilDataSearch(cari, pos);
55                break;
56            case 7:
57                dsn.SortingASC(); // wajib urut ASC sebelum binary search
58                System.out.print(s:"Masukkan usia dosen yang dicari: ");
59                int cariUsia = sc.nextInt();
60                int posUsia = dsn.binarySearchDenganPeringatan(cariUsia);
61                if (posUsia != -1) {
62                    System.out.println("Data dosen dengan usia " + cariUsia + " ditemukan pada indeks ke-" + posUsia);
63                    dsn.tampilDataSearch(dsn.data11[posUsia].nama, posUsia);
64                } else {
65                    System.out.println("Data dosen dengan usia " + cariUsia + " tidak ditemukan.");
66                }
67                break;
68            case 8:
69                System.out.println(x:"Keluar dari program.");
70                break;
71            default:
72                System.out.println(x:"Pilihan tidak valid!");
73        }
74    }
75    } while (pilihan != 8);

```

```
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Sequential Search dengan Nama Dosen
7. Binary Search dengan Usia Dosen (ASC)
8. Keluar
Pilih menu: 6
Masukkan nama dosen yang dicari: budi
Peringatan: Ditemukan lebih dari satu dosen dengan nama yang sama!
Data dosen dengan nama: budi ditemukan pada index ke-0
kode      : 111
nama      : budi
jenis kelamin : Laki-laki
usia      : 23

Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Sequential Search dengan Nama Dosen
7. Binary Search dengan Usia Dosen (ASC)
8. Keluar
Pilih menu: 7
Data berhasil diurutkan secara ascending (usia termuda ke tertua).
Masukkan usia dosen yang dicari: 21
Peringatan: Ditemukan lebih dari satu dosen dengan usia yang sama!
Data dosen dengan usia 21 ditemukan pada indeks ke-0
kode      : 222
nama      : santi
jenis kelamin : Perempuan
usia      : 21
```

Menu:

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Sequential Search dengan Nama Dosen
7. Binary Search dengan Usia Dosen (ASC)
8. Keluar

Pilih menu: 2

Kode: 222

Nama: santi

Jenis Kelamin: Perempuan

Usia: 21

Kode: 444

Nama: rati

Jenis Kelamin: Perempuan

Usia: 21

Kode: 111

Nama: budi

Jenis Kelamin: Laki-laki

Usia: 23

Kode: 333

Nama: asep

Jenis Kelamin: Laki-laki

Usia: 25

Kode: 555

Nama: budi

Jenis Kelamin: Laki-laki

Usia: 30
