

LAPORAN HASIL PRAKTIKUM

ALGORITMA dan STRUKTUR DATA

JOBSHEET 7



Alexsa Fitria Ayu Siswoyo.

244107020020

TI - 1E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

PERCOBAAN 1

Langkah-langkah Percobaan Sequential Search

1. Pada pertemuan Jobsheet 7 ini akan menggunakan class Mahasiswa<no Presensi>, MahasiswaBerprestasi<no Presensi>, dan MahasiswaDemo<no presensi> 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<no presensi>. 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<no presensi>.

```
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");  
    }  
}
```

5. Buatlah method tampilPoisisi bertipe void dan Deklarasikan isi dari method tampilPoisisi pada class MahasiswaBerprestasi<no presensi>.

```
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<noPresensi>, tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.

```
package jobsheet7;

import java.util.Scanner;
public class MahasiswaDemo02 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi02 list = new
        MahasiswaBerprestasi02(5);
        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 Mahasiswa02(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();

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

```

7. jalankan dan amati hasilnya

```

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
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:964)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextDouble(Scanner.java:2590)
    at jobsheet7.MahasiswaDemo02.main(MahasiswaDemo02.java:32)
```

PERTANYAAN

1. Jelaskan perbedaan method tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!
 - Fungsi tampilDataSearch adalah untuk mencari data dalam suatu kumpulan berdasarkan kriteria yang diberikan, kemudian menampilkan hasil pencarian tersebut.
 - Fungsi tampilPosisi adalah untuk menampilkan posisi suatu data dalam kumpulan data dan memberikan informasi mengenai lokasi fisik atau logis data tersebut dalam penyimpanan.
2. elaskan fungsi break pada kode program dibawah ini!

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

- Jika nilai IPK pada elemen listMhs[j] dalam kode program sama dengan nilai cari, maka variabel posisi akan diisi dengan nilai indeks j dan perulangan akan dihentikan oleh pernyataan break, sehingga program tidak perlu lagi memeriksa elemen berikutnya dalam array.

PERCOBAAN 2

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 `tampilPoisisi` dan `tampilDataSearch`.

```
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("-----
-----");
int posisiBinary =
list.findBinarySearch(cari, 0, jumMhs - 1);
list.tampilPosisi(cari, posisiBinary);
list.tampilDataSearch(cari, posisiBinary);
```

3. Jalankan dan amati hasilnya

```
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
```

```
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:964)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextDouble(Scanner.java:2590)
    at jobsheet7.MahasiswaDemo02.main(MahasiswaDemo02.java:32)
```

```
PS C:\Users\User\OneDrive\KAMPUS\KELAS\1E\SEM 2\Praktikum-ASD>
```

PERTANYAAN

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

- proses divide dilakukan oleh

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

- Ruang pencarian dibagi dua setelah nilai tengah (mid) dihitung; pencarian dilanjutkan ke kiri (findBinarySearch(cari, left, mid - 1)) jika nilai yang dicari lebih kecil, dan ke kanan (findBinarySearch(cari, mid + 1, right)) jika lebih besar

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

- Ketika elemen yang dicari (cari) identik dengan IPK pada listMhs[mid], fungsi akan mengembalikan mid, menunjukkan keberhasilan pencarian.

```
if (cari == listMhs[mid].ipk) {  
    return mid;  
}
```

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

- Algoritma Binary Search tidak akan berfungsi dengan semestinya jika data IPK yang diberikan tidak dalam keadaan terurut. Hal ini disebabkan karena cara kerja Binary Search yang sangat bergantung pada susunan data terurut untuk membagi area pencarian menjadi dua bagian secara logis. Akibatnya, indeks tengah (mid) yang terhitung tidak merepresentasikan posisi relatif elemen yang lebih besar atau lebih kecil dalam keseluruhan data. Konsekuensinya, proses pencarian elemen dapat menghasilkan hasil yang salah atau bahkan gagal menemukan elemen yang sebenarnya ada di dalam daftar data. Oleh karena itu, untuk menjamin algoritma Binary Search dapat berjalan dengan benar, data perlu diurutkan terlebih dahulu. Proses pengurutan ini dapat dilakukan menggunakan berbagai metode seperti Bubble Sort, Selection Sort, atau Insertion Sort sebelum fungsi Binary Search dipanggil untuk melakukan pencarian.

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.

- Karena algoritma binary search dalam kode program dirancang untuk data yang diurutkan secara menaik (ascending), bukan menurun (descending), maka pengurutan IPK dari terbesar ke terkecil akan mengakibatkan hasil yang tidak sesuai

- Ubah perbandingan kode program

```
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) { //ubah
perbandingan
            return findBinarySearch(cari, left, mid -
1);
        } else {
            return findBinarySearch(cari, mid + 1,
right);
        }
    }
    return -1;
}
```

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

- Modifikasi pada class MahasiswaDemo

```
Scanner sc = new Scanner(System.in);
System.out.print("Masukkan jumlah mahasiswa: ");
int jumMhs = sc.nextInt();
MahasiswaBerprestasi02 list = new
MahasiswaBerprestasi02(jumMhs);

for (int i = 0; i < jumMhs; i++) {
    System.out.println("Masukkan Data Mahasiswa
ke-" + (i + 1));
    System.out.print("NIM    : ");
    String nim = sc.next();
    System.out.print("Nama    : ");
    String nama = sc.next();
    System.out.print("Kelas : ");
    String kelas = sc.next();
    System.out.print("IPK    : ");
    String ipk = sc.next();
    Double ipk = Double.parseDouble(ipk);
    sc.nextLine();
    System.out.println("-----
-----");
    Mahasiswa02 mhs = new Mahasiswa02(nim, nama,
kelas, ipk);
    list.tambah(mhs);
}
```

- Outputnya

```
Masukkan jumlah mahasiswa: 3
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 : 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 : fia
NIM : 333
Kelas : 2
IPK : 3.3
-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari :
IPK: 3.8
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:964)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextDouble(Scanner.java:2590)
    at jobsheet7.MahasiswaDemo02.main(MahasiswaDemo02.java:36)
PS C:\Users\User\OneDrive\KAMPUS\KELAS\1E\SEM 2\Praktikum-ASD>
```

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:

- a. PencarianDataSequential<no presensi>: digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.

```
//A.
void pencarianDataSequential02(String nama) {
    int jumlahHasil = 0;
    for (int i = 0; i < idx; i++) {
        if
(DataDosen02[i].nama.equalsIgnoreCase(nama)) {
            DataDosen02[i].tampil();
            jumlahHasil++;
        }
    }
    if (jumlahHasil > 1) {
        System.out.println("Peringatan! Pastikan
algoritma yang diterapkan sesuai dengan kasus yang
diberikan!");
    } else if (jumlahHasil == 0) {
        System.out.println("Data tidak ditemukan.");
    }
}
```

- b. encarianDataBinary<no presensi>: digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.

```
//B.
void pencarianDataBinary02(int usia) {
    sortingASC();
    int kiri = 0, kanan = idx - 1;
    boolean ditemukan = false;

    while (kiri <= kanan) {
        int tengah = (kiri + kanan) / 2;
        if (DataDosen02[tengah].usia == usia) {
            DataDosen02[tengah].tampil();
            ditemukan = true;
            break;
        } else if (DataDosen02[tengah].usia < usia) {
            kiri = tengah + 1;
        } else {
            kanan = tengah - 1;
        }
    }
    if (!ditemukan) {
        System.out.println("Data tidak ditemukan.");
    }
}
```

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

```
while (pilihan != 7);  
    System.out.println("Peringatan! Pastikan  
    algoritma yang diterapkan sesuai dengan kasus yang  
    diberikan!");
```

- Kode program Dosen<no presensi>

```
package jobsheet7;  
  
public class Dosen02 {  
    String kode;  
    String nama;  
    boolean jenisKelamin;  
    int usia;  
  
    Dosen02 (String kd, String name, boolean jk, int  
age) {  
        kode = kd;  
        nama = name;  
        jenisKelamin = jk;  
        usia = age;  
    }  
  
    void tampil() {  
        System.out.println("Kode : " + kode);  
        System.out.println("Nama : " + nama);  
        System.out.println("Jenis Kelamin : " +  
jenisKelamin);  
        System.out.println("Usia : " + usia);  
    }  
}
```

- Kode program DataDosen<no presensi>

```
package jobsheet7;

public class DataDosen02 {
    Dosen02[] DataDosen02 = new Dosen02[2];
    int idx;

    void tambah(Dosen02 dsn) {
        if (idx < DataDosen02.length) {
            DataDosen02[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data Dosen Sudah
Penuh!");
        }
    }

    //menampilkan data dosen
    void tampil() {
        if (idx == 0) {
            System.out.println("Tidak Ada Data Dosen");
            return;
        }
        for (int i = 0; i < idx; i++) {
            DataDosen02[i].tampil();
            System.out.println();
        }
    }

    //A.
    void pencarianDataSequential02(String nama) {
        int jumlahHasil = 0;
        for (int i = 0; i < idx; i++) {
            if
(DataDosen02[i].nama.equalsIgnoreCase(nama)) {
                DataDosen02[i].tampil();
                jumlahHasil++;
            }
        }
        if (jumlahHasil > 1) {
            System.out.println("Peringatan! Pastikan
algoritma yang diterapkan sesuai dengan kasus yang
diberikan!");
        } else if (jumlahHasil == 0) {
            System.out.println("Data tidak ditemukan.");
        }
    }
}
```

```

//B.
void pencarianDataBinary02(int usia) {
    sortingASC();
    int kiri = 0, kanan = idx - 1;
    boolean ditemukan = false;

    while (kiri <= kanan) {
        int tengah = (kiri + kanan) / 2;
        if (DataDosen02[tengah].usia == usia) {
            DataDosen02[tengah].tampil();
            ditemukan = true;
            break;
        } else if (DataDosen02[tengah].usia < usia) {
            kiri = tengah + 1;
        } else {
            kanan = tengah - 1;
        }
    }
    if (!ditemukan) {
        System.out.println("Data tidak ditemukan.");
    }
}

//mengurutkan data dosen secara ascending (asc)
void sortingASC() {
    for (int i = 0; i < idx - 1; i++) {
        for (int j = 0; j < idx - 1 - i; j++) {
            if (DataDosen02[j].usia >
DataDosen02[j+1].usia) {
                Dosen02 temp = DataDosen02[j];
                DataDosen02[j] = DataDosen02[j+1];
                DataDosen02[j+1] = temp;
            }
        }
    }
}

//mengurutkan data dosen secara descending (dsc)
void sortingDSC() {
    for (int i = 0; i < idx - 1; i++) {
        for (int j = 0; j < idx - 1 - i; j++) {
            if (DataDosen02[j].usia <
DataDosen02[j+1].usia) {
                Dosen02 temp = DataDosen02[j];
                DataDosen02[j] = DataDosen02[j+1];
                DataDosen02[j+1] = temp;
            }
        }
    }
}
}

```

- Kode program DosenDemo<no presenssi>

```
package jobsheet7;

import java.util.Scanner;
public class DosenDemo02 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen02 datadsn = new DataDosen02();
        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 ASC (Usia
Termuda ke Tertua)");
            System.out.println("4. Sorting DSC (Usia
Tertua ke Termuda)");
            System.out.println("5. Mencari Nama Dosen");
            System.out.println("6. Mencari Usia Dosen");
            System.out.println("7. Keluar");
            System.out.print("\nPilih Menu : ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    for (int i = 0; i < 3; i++) {
                        System.out.println("Masukkan data
dosen ke-" + (i+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 (P / L) : ");
                        char jk = sc.next().charAt(0);
                        boolean jenisKelamin = (jk == '1'
|| jk == 'p');
                        System.out.print("Masukkan usia
dosen: ");
                        int usia = sc.nextInt();
                        sc.nextLine();
                    }
                }
            }
        } while (pilihan != 7);
    }
}
```

```

        sc.nextLine();
        System.out.println("-----
-----");

        Dosen02 dsn = new Dosen02(kode,
nama, jenisKelamin, usia);
        datadsn.tambah(dsn);
    }
    break;
case 2:
    datadsn.tampil();
    break;
case 3:
    datadsn.sortingASC();
    System.out.println("Data Dosen Telah
Diurutkan Secara Ascending");
    datadsn.tampil();
    break;
case 4:
    datadsn.sortingDSC();
    System.out.println("Data Dosen Telah
Diurutkan Secara Descending");
    datadsn.tampil();
    break;
case 5:
    System.out.print("Masukkan nama dosen
yang dicari: ");
    String namaCari = sc.nextLine();
    datadsn.pencarianDataSequential02(nam
aCari);
    break;
case 6:
    System.out.print("Masukkan usia dosen
yang dicari: ");
    int usiaCari = sc.nextInt();
    sc.nextLine();
    datadsn.pencarianDataBinary02(usiaCar
i);
    break;
default:
    System.out.println("Pilihan Tidak
Valid. Silahkan Coba Lagi.");
}
}
while (pilihan != 7);
System.out.println("Peringatan! Pastikan
algoritma yang diterapkan sesuai dengan kasus yang
diberikan!");
}
}

```


- Outputnya menu 1

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 1
Masukkan data dosen ke-1:
Masukkan kode dosen: 111
Masukkan nama dosen: adi
Masukkan jenis Kelamin (P / L) : 1
Masukkan usia dosen: 32
-----
Masukkan data dosen ke-2:
Masukkan kode dosen: 222
Masukkan nama dosen: ida
Masukkan jenis Kelamin (P / L) : p
Masukkan usia dosen: 43
-----
Masukkan data dosen ke-3:
Masukkan kode dosen: 333
Masukkan nama dosen: dai
Masukkan jenis Kelamin (P / L) : 1
Masukkan usia dosen: 29
-----
Data Dosen Sudah Penuh!
```

- Outputnya menu 2

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 2
Kode : 111
Nama : adi
Jenis Kelamin : false
Usia : 32

Kode : 222
Nama : ida
Jenis Kelamin : true
Usia : 43
```

- Outputnya menu 3

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 3
Data Dosen Telah Diurutkan Secara Ascending
Kode : 111
Nama : adi
Jenis Kelamin : false
Usia : 32

Kode : 222
Nama : ida
Jenis Kelamin : true
Usia : 43
```

- Outputnya menu 4

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 4
Data Dosen Telah Diurutkan Secara Descending
Kode : 222
Nama : ida
Jenis Kelamin : true
Usia : 43

Kode : 111
Nama : adi
Jenis Kelamin : false
Usia : 32
```

- Outputnya menu 5

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 5
Masukkan nama dosen yang dicari: ida
Kode : 222
Nama : ida
Jenis Kelamin : true
Usia : 43
```

- Outputnya menu 6

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 6
Masukkan usia dosen yang dicari: 32
Kode : 111
Nama : adi
Jenis Kelamin : false
Usia : 32
```

- Outputnya menu 7

```
Menu
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Termuda ke Tertua)
4. Sorting DSC (Usia Tertua ke Termuda)
5. Mencari Nama Dosen
6. Mencari Usia Dosen
7. Keluar

Pilih Menu : 7
Pilihan Tidak Valid. Silahkan Coba Lagi.
Peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!
PS C:\Users\User\OneDrive\KAMPUS\KELAS\1E\SEM 2\Praktikum-ASD> █
```

