

LAPORAN HASIL PRAKTIKUM ALSD
JOBSHEET 7 SEARCHING



Faiva Puspa Sahara

244107020036

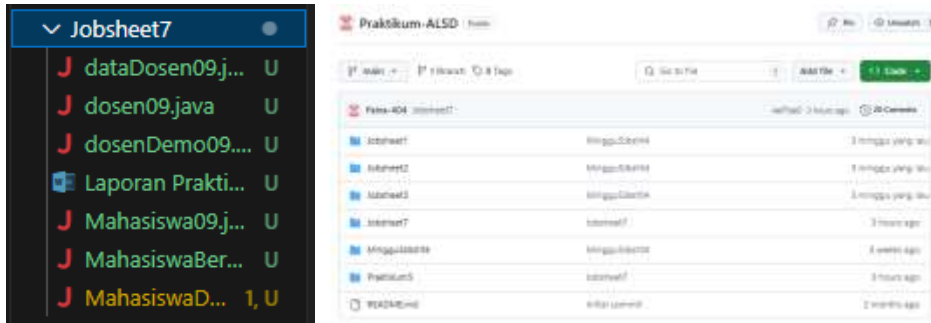
TI – 1E

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

HASIL PRAKTIKUM

7.2 Percobaan 1 (Searching – Sequential Search)

1. Buat folder baru bernama **Jobsheet7**, kemudian buka ketiga class dari Jobsheet sebelumnya dan copy ke folder Jobsheet 7



2. Tambahkan method **sequentialSearching** bertipe integer dengan parameter **cari** bertipe double. Buatlah method **tampilPosisi** bertipe void dan deklarasikan isi dari method. Buatlah method **tampilDataSearch** bertipe void dan deklarasikan isi dari method. Pada class **MahasiswaBerprestasi09**.

```
//sequentialSearching
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;
}

//tampilPosisi
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");
    }
}

//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");
    }
}
```

3. Pada class **MahasiswaDemo09** , tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.

```
package Jobsheet7;

import java.util.Scanner;
public class MahasiswaDemo09 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi09 list = new
MahasiswaBerprestasi09();
        int jumMhs = 5;

        for (int i = 0; i < jumMhs; i++) {
            System.out.println("Masukkan Data Mahasiswa ke-" + (i
+ 1));

            System.out.print("NIM    : ");
            String nim = sc.nextLine();
            System.out.print("Nama    : ");
            String nama = sc.nextLine();
            System.out.print("Kelas : ");
            String kelas = sc.nextLine();
            System.out.print("IPK    : ");
            String ipk = sc.nextLine();
            Double ipk = Double.parseDouble(ipk);
            System.out.println("-----
---");

            list.tambah(new Mahasiswa09(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.println("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);

    }
}
```

4. Run 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:\SAHARA\KULIAH\SEM 2\Praktikum-ALSD>
```

➤ Pertanyaan

1. Jelaskan perbedaan metod **tampilDataSearch** dan **tampilPosisi** pada class MahasiswaBerprestasi09!
 - ⇒ **tampilDataSearch** : Berfungsi untuk (mencari dan menampilkan data) berdasarkan kriteria, melibatkan proses pencarian dalam suatu kumpulan data, dan menampilkan hasil pencarian yang sesuai dengan kriteria.
 - ⇒ **tampilPosisi** : Berfungsi untuk (menampilkan posisi data tertentu) dalam kumpulan data dan menginformasikan lokasi fisik atau logis dari data dalam struktur penyimpanan.

2. Jelaskan fungsi break pada kode program berikut!

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

- ⇒ Jika dalam kode program nilai IPK pada elemen listMhs[j] sama dengan nilai cari, maka variabel posisi diisi dengan nilai indeks j.
- ⇒ Setelah kondisi terpenuhi, pernyataan break menghentikan perulangan, sehingga program tidak perlu terus memeriksa elemen berikutnya dalam array.

7.3 Percobaan 2 (Searching – Binary Search)

1. Tambahkan method **findBinarySearch** bertipe int pada class **MahasiswaBerprestasi09**. 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 **MahasiswaBerprestasi09** di kelas **MahasiswaDemo09**. Kemudian panggil method **tampilPosisi** dan **tampilDataSearch**.

```
//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 sequential searching");  
System.out.println("-----  
");  
double posisi2 = list.findBinarySearch(cari, 0, jumMhs-1);  
int pss2 = (int)posisi2;  
list.tampilPosisi(cari, pss2);  
list.tampilDataSearch(cari, pss2);  
}
```

3. Run 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 dicari :
IPK : 3,7
-----
menggunakan sequential searching
-----
data mahasiswa dengan IPK :3.7 ditemukan pada indeks 4
nim      : 555
nama     : Anita
kelas    : 2
ipk      : 3.7
```

➤ Pertanyaan

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

⇒ Proses **divide** dilakukan pada

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

⇒ Setelah nilai tengah (`mid`) dihitung, ruang pencarian dibagi menjadi dua. Jika nilai yang dicari lebih kecil, pencarian dilanjutkan ke kiri `findBinarySearch(cari, left, mid - 1)`. Jika lebih besar, pencarian ke kanan `findBinarySearch(cari, mid + 1, right)`.

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

⇒ Jika elemen yang dicari (`cari`) sama dengan elemen di indeks tengah (`listMhs[mid].ipk`), maka indeks tersebut dikembalikan (`return (mid)`), yang menandakan proses pencarian berhasil.

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

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

⇒ Tidak akan berjalan dengan benar jika data IPK yang dimasukkan **tidak urut**. Karena algoritma Binary Search **mengandalkan data yang sudah terurut** untuk membagi ruang pencarian menjadi dua bagian secara logis.

⇒ Indeks tengah (*mid*) yang dihitung tidak mencerminkan posisi elemen yang lebih besar atau lebih kecil dalam data. Akibatnya, pencarian elemen menghasilkan salah atau gagal menemukan elemen yang sebenarnya ada dalam daftar.

⇒ Untuk memastikan algoritma Binary Search berjalan benar, data perlu diurutkan dahulu menggunakan metode pengurutan **bubbleSort**, **selectionSort**, atau **insertionSort** sebelum memanggil fungsi pencarian Binary Search.

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.

⇒ Jika IPK diurutkan dari terbesar ke terkecil, **binary search dalam kode program tidak memberikan hasil yang sesuai**. Algoritma binary search dalam kode dirancang untuk bekerja dengan data yang diurutkan secara **menaik (ascending)**, bukan **menurun (descending)**.

⇒ Ubah 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.

⇒ Pada class **MahasiswaBerprestasi09** modifikasi kode program

```
public class MahasiswaBerprestasi09 {
    Mahasiswa09[] listMhs;
    int idx;

    MahasiswaBerprestasi09(int jumMhs) {
        listMhs = new Mahasiswa09[jumMhs];
        idx = 0;
    }
}
```

⇒ Pada class **MahasiswaDemo09** modifikasi kode program

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Masukkan jumlah mahasiswa: ");
    int jumMhs = sc.nextInt();
    MahasiswaBerprestasi09 list = new
MahasiswaBerprestasi09(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("-----
-----");
        Mahasiswa09 mhs = new Mahasiswa09(nim, nama,
kelas, ipk);
        list.tambah(mhs);
    }
}
```

⇒ Run program

```
Masukkan jumlah mahasiswa: 2
Masukkan Data Mahasiswa ke-1
NIM    : 111
Nama    : Adi
Kelas  : 2
IPK     : 3.1
-----
Masukkan Data Mahasiswa ke-2
NIM    : 222
Nama    : Ila
Kelas  : 2
IPK     : 3.2
-----
Nama    : Adi
NIM    : 111
Kelas  : 2
IPK     : 3.1
-----
Nama    : Ila
NIM    : 222
Kelas  : 2
IPK     : 3.2
-----
```

```
-----
Pencarian data
-----
masukkan ipk mahasiswa yang dicari :
IPK : 3,1
-----
Menggunakan sequential searching
-----
data mahasiswa dengan IPK :3.1 ditemukan pada indeks 0
nim      : 111
nama     : Adi
kelas    : 2
ipk      : 3.1
PS D:\SAHARA\KULIAH\SEM 2\Praktikum-ALSD> █
```

6. Commit dan push

```
PS D:\SAHARA\KULIAH\SEM 2\Praktikum-ALSD> git add .
PS D:\SAHARA\KULIAH\SEM 2\Praktikum-ALSD> git commit -m "08 April"
[main 6c626a5] 08 April
3 files changed, 21 insertions(+), 13 deletions(-)
delete mode 100644 Jobsheet7/~$poran Praktikum.docx
PS D:\SAHARA\KULIAH\SEM 2\Praktikum-ALSD> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 681 bytes | 136.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/Faiva-404/Praktikum-ALSD.git
1fa4238..6c626a5 main -> main
PS D:\SAHARA\KULIAH\SEM 2\Praktikum-ALSD> █
```


7.5 Latihan Praktikum

1. Pada Latihan praktikum pertemuan sebelumnya pada Jobsheet 6 yang terdapat 3 class yaitu **Dosen09**, **DataDosen09**, dan **DosenDemo09**, tambahkan method:

a. Pencarian **DataSequential09** : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma **sequential search**.

```
void pencarianDataSequential09(String nama) {
    int jumlahHasil = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen09[i].nama.equalsIgnoreCase(nama)) {
            dataDosen09[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. Pencarian **DataBinary09**: digunakan untuk mencari data dosen berdasarkan usia dengan algoritma **binary search**.

```
void pencarianDataBinary09(int usia) {
    sortingASC();
    int kiri = 0, kanan = idx - 1;
    boolean ditemukan = false;

    while (kiri <= kanan) {
        int tengah = (kiri + kanan) / 2;
        if (dataDosen09[tengah].usia == usia) {
            dataDosen09[tengah].tampil();
            ditemukan = true;
            break;
        } else if (dataDosen09[tengah].usia < usia) {
            kiri = tengah + 1;
        } else {
            kanan = tengah - 1;
        }
    }
    if (!ditemukan) {
        System.out.println("Data tidak ditemukan.");
    }
}
```

c. Buat 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 dosen09.java

```
package Jobsheet7;

public class dosen09 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    dosen09 (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 dataDosen09.java

```
package Jobsheet7;

public class dataDosen09 {
    dosen09[] dataDosen09 = new dosen09[9];
    int idx;

    void tambah(dosen09 dsn) {
        if (idx < dataDosen09.length) {
            dataDosen09[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++) {
            dataDosen09[i].tampil();
            System.out.println();
        }
    }

    //A.
    void pencarianDataSequential09(String nama) {
        int jumlahHasil = 0;
        for (int i = 0; i < idx; i++) {
            if (dataDosen09[i].nama.equalsIgnoreCase(nama)) {
                dataDosen09[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 pencarianDataBinary09(int usia) {
        sortingASC();
        int kiri = 0, kanan = idx - 1;
        boolean ditemukan = false;

        while (kiri <= kanan) {
            int tengah = (kiri + kanan) / 2;
            if (dataDosen09[tengah].usia == usia) {
                dataDosen09[tengah].tampil();
                ditemukan = true;
                break;
            } else if (dataDosen09[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 (dataDosen09[j].usia > dataDosen09[j+1].usia)
                {
                    dosen09 temp = dataDosen09[j];
                    dataDosen09[j] = dataDosen09[j+1];
                    dataDosen09[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 (dataDosen09[j].usia < dataDosen09[j+1].usia)
                {
                    dosen09 temp = dataDosen09[j];
                    dataDosen09[j] = dataDosen09[j+1];
                    dataDosen09[j+1] = temp;
                }
            }
        }
    }
}

```

⇒ Kode program **dataDemo09.java**

```

package Jobsheet7;

import java.util.Scanner;
public class dosenDemo09 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        dataDosen09 datadsn = new dataDosen09();
        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 < 10; 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();
                    System.out.println("-----
-----");

                    dosen09 dsn = new dosen09(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.pencarianDataSequential09(namaCari);
                break;
            case 6:
                System.out.print("Masukkan usia dosen yang
dicari: ");
                int usiaCari = sc.nextInt();
                sc.nextLine();
                datadsn.pencarianDataBinary09(usiaCari);
                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!");
}
}

```

⇒ Output 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: Aldi
Masukkan jenis Kelamin (P / L) : L
Masukkan usia dosen: 28
-----
Masukkan data dosen ke-2:
Masukkan kode dosen: 222
Masukkan nama dosen: Dyo
Masukkan jenis Kelamin (P / L) : L
Masukkan usia dosen: 49
-----
Masukkan data dosen ke-3:
Masukkan kode dosen: 333
Masukkan nama dosen: Dinar
Masukkan jenis Kelamin (P / L) : P
Masukkan usia dosen: 27
-----
Masukkan data dosen ke-4:
Masukkan kode dosen: 444
Masukkan nama dosen: Dinar
Masukkan jenis Kelamin (P / L) : L
Masukkan usia dosen: 36
-----
Masukkan data dosen ke-5:
Masukkan kode dosen: 555
Masukkan nama dosen: Rafi
Masukkan jenis Kelamin (P / L) : L
Masukkan usia dosen: 30
-----
Masukkan data dosen ke-6:
Masukkan kode dosen: 666
Masukkan nama dosen: Ara
Masukkan jenis Kelamin (P / L) : P
Masukkan usia dosen: 28
-----
Masukkan data dosen ke-7:
Masukkan kode dosen: 777
Masukkan nama dosen: Putra
Masukkan jenis Kelamin (P / L) : L
Masukkan usia dosen: 31
-----
Masukkan data dosen ke-8:
Masukkan kode dosen: 888
Masukkan nama dosen: Putri
Masukkan jenis Kelamin (P / L) : P
Masukkan usia dosen: 25
-----
Masukkan data dosen ke-9:
Masukkan kode dosen: 999
Masukkan nama dosen: Ujang
Masukkan jenis Kelamin (P / L) : L
Masukkan usia dosen: 41
-----
Masukkan data dosen ke-10:
Masukkan kode dosen: 100
Masukkan nama dosen: Luyna
Masukkan jenis Kelamin (P / L) : P
Masukkan usia dosen: 55
-----
Data Dosen Sudah Penuh!

```

⇒ Output 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 : Aldi
Jenis Kelamin : false
Usia : 28

Kode : 222
Nama : Dyo
Jenis Kelamin : false
Usia : 49

Kode : 333
Nama : Dinar
Jenis Kelamin : false
Usia : 27

Kode : 444
Nama : Dinar
Jenis Kelamin : false
Usia : 36

Kode : 555
Nama : Rafi
Jenis Kelamin : false
Usia : 30

Kode : 666
Nama : Ara
Jenis Kelamin : false
Usia : 28

Kode : 777
Nama : Putra
Jenis Kelamin : false
Usia : 31

Kode : 888
Nama : Putri
Jenis Kelamin : false
Usia : 25

Kode : 999
Nama : Ujang
Jenis Kelamin : false
Usia : 41

```

⇒ Output 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 : 888
Nama : Putri
Jenis Kelamin : false
Usia : 25

Kode : 333
Nama : Dinar
Jenis Kelamin : false
Usia : 27

Kode : 111
Nama : Aldi
Jenis Kelamin : false
Usia : 28

Kode : 666
Nama : Ara
Jenis Kelamin : false
Usia : 28

Kode : 555
Nama : Rafi
Jenis Kelamin : false
Usia : 30

Kode : 777
Nama : Putra
Jenis Kelamin : false
Usia : 31

Kode : 444
Nama : Dinar
Jenis Kelamin : false
Usia : 36

Kode : 999
Nama : Ujang
Jenis Kelamin : false
Usia : 41

Kode : 222
Nama : Dyo
Jenis Kelamin : false
Usia : 49
```

⇒ Output 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 : Dyo
Jenis Kelamin : false
Usia : 49

Kode : 999
Nama : Ujang
Jenis Kelamin : false
Usia : 41

Kode : 444
Nama : Dinar
Jenis Kelamin : false
Usia : 36

Kode : 777
Nama : Putra
Jenis Kelamin : false
Usia : 31

Kode : 555
Nama : Rafi
Jenis Kelamin : false
Usia : 30

Kode : 111
Nama : Aldi
Jenis Kelamin : false
Usia : 28

Kode : 666
Nama : Ara
Jenis Kelamin : false
Usia : 28

Kode : 333
Nama : Dinar
Jenis Kelamin : false
Usia : 27

Kode : 888
Nama : Putri
Jenis Kelamin : false
Usia : 25
```

⇒ Output Menu 5

```
Pilih Menu : 5
Masukkan nama dosen yang dicari: Dinar
Kode : 444
Nama : Dinar
Jenis Kelamin : false
Usia : 36
Kode : 333
Nama : Dinar
Jenis Kelamin : false
Usia : 27
Peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

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: Ara
Kode : 666
Nama : Ara
Jenis Kelamin : false
Usia : 28
```

⇒ Output Menu 6

```
Pilih Menu : 6
Masukkan usia dosen yang dicari: 31
Kode : 777
Nama : Putra
Jenis Kelamin : false
Usia : 31

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: 60
Data tidak ditemukan.

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
```

⇒ Output 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 : 8
Pilihan Tidak Valid. Silahkan Coba Lagi.









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 D:\SAWARA\KULIAH\SEM 2\Praktikum-ALSD>
```

⇒ Commit dan Push

Praktikum-ALSD / JobSheet7 / 

File 434 (18 Apr) add file 1 minutes ago History

Name	Last commit message	Last commit date
 ..		
 Laporan Praktikum.docx	07 April	yesterday
 Mahasiswa09.java	10/04/2021	5 days ago
 MahasiswaBerprestasi09.java	18 April	2 hours ago
 MahasiswaCero09.java	18 April	2 hours ago
 DataDosen09.java	18 April	3 minutes ago
 Dosen09.java	10/04/2021	5 days ago
 DosenDemo09.java	18 April	5 minutes ago