

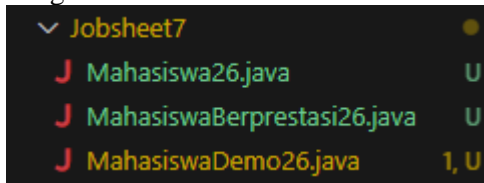
LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 7



RIFO ANGGI BARBARA DANUARTA
244107020063
TI_1E
PROGRAM STUDI D_IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Percobaan 1

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 indeks " + 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.

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    input.useLocale(Locale.US);
    int jumMhs = 5;

    MahasiswaBerprestasi26 list = new
MahasiswaBerprestasi26();

    for (int i = 0; i < jumMhs; i++ ) {
        System.out.println("Masukan Data mahasiswa ke-" + (i +
1));
        System.out.print("Nim    : ");
        String nim = input.nextLine();
        System.out.print("Nama    : ");
        String nama = input.nextLine();
        System.out.print("Kelas : ");
        String kelas = input.nextLine();
        System.out.print("IPK     : ");
        double IPK = input.nextDouble();
        input.nextLine();

        Mahasiswa26 m = new Mahasiswa26(nim, nama, kelas,
IPK);
        list.tambah(m);
        System.out.println("-----");
    }
    list.tampil();
    System.out.println("-----");
    System.out.println("Pencarian data");
    System.out.println("-----");
    System.out.println("masukan ipk mahasiswa yang dicari:");
    System.out.print("IPK: ");
    double cari = input.nextDouble();

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

7. Jalankan dan amati hasilnya.

```

Masukan Data mahasiswa ke-1
Nim : 111
Nama : adi
Kelas : 2
IPK : 3.6
-----
Masukan Data mahasiswa ke-2
Nim : 222
Nama : tio
Kelas : 2
IPK : 3.8
-----
Masukan Data mahasiswa ke-3
Nim : 333
Nama : ila
Kelas : 2
IPK : 3.0
-----
Masukan Data mahasiswa ke-4
Nim : 444
Nama : lia
Kelas : 2
IPK : 3.5
-----
Masukan 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
-----
masukan 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

```

Pertanyaan

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!
2. Jelaskan fungsi break pada kode program dibawah ini!

```

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

```

Jawaban

1. Method tampilPosisi = menampilkan apakah IPK ada dalam data dan menampilkan posisi IPK yang dicari pada indeks ke berapa

Method tampilDataSearch = menampilkan data nim,nama,kelas jika IPK yang dicari ditemukan

2. Yaitu untuk menghentikan loop agar tidak mencari ke indeks selanjutnya
Fungsi agar efisien, menghemat waktu eksekusi, dan menghindari perubahan yang tidak diperlukan

Percobaan 2

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`

```
System.out.println("-----");
-----");
System.out.println("Pencarian data");
System.out.println("-----");
-----");
System.out.println("masukan ipk mahasiswa yang dicari:");
");
System.out.print("IPK: ");
double caril = input.nextDouble();
System.out.println("-----");
System.out.println("menggunakan binary search");
System.out.println("-----");
double posisi2 = list.findBinarySearch(caril, 0,
jumMhs-1);
int pss2 = (int) posisi2;
list.tampilPosisi(caril, pss2);
list.tampilDataSearch(caril, pss2);
System.out.println();
```

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

```
Masukan Data mahasiswa ke-1
Nim   : 111
Nama  : adi
Kelas : 2
IPK   : 3.1
-----
Masukan Data mahasiswa ke-2
Nim   : 222
Nama  : ila
Kelas : 2
IPK   : 3.2
-----
Masukan Data mahasiswa ke-3
Nim   : 333
Nama  : lia
Kelas : 2
IPK   : 3.3
-----
Masukan Data mahasiswa ke-4
Nim   : 444
Nama  : susi
Kelas : 2
IPK   : 3.5
-----
Masukan Data mahasiswa ke-5
Nim   : 555
Nama  : anita
Kelas : 2
IPK   : 3.7
-----
-----
Pencarian data
-----
masukan 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
```

Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
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
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

Jawaban

1. $mid = (left + right) / 2$; karena setiap kali rekursif dipanggil array dibagi 2 dan proses ini berulang sampai data ditemukan.
2. `if (cari1 == listMhs[mid].ipk) return mid`; karena setelah elemen yang dicari ditemukan dan langsung mengembalikan hasilnya.
3. Masih bisa berjalan tapi hasilnya tidak akan akurat karena binary search menganggap bahwa semua elemen disebelah kiri lebih kecil dan semua elemen disebelah kanan lebih besar, dan solusinya adalah sebelum melakukan binary search data harus sudah terurut dulu.
4. Hasilnya tidak akan sesuai atau tidak ditemukan

```
int findBinarySearch(double cari1, int left, int right) {
    boolean ascending = listMhs[left].ipk < listMhs[right].ipk;

    while (left <= right) {
        int mid = (left + right) / 2;

        if (listMhs[mid].ipk == cari1) {
            return mid;
        }

        if (ascending) {
            if (listMhs[mid].ipk > cari1) {
                right = mid - 1;
            } else {
                left = mid + 1;
            }
        } else {
            if (listMhs[mid].ipk < cari1) {
                right = mid - 1;
            } else {
                left = mid + 1;
            }
        }
    }
    return -1;
}
```

5.

```
System.out.print("Masukan Jumlah mahasiswa: ");
int jlmMhs = input.nextInt();
input.nextLine();

for (int i = 0; i < jlmMhs; i++ ) {
    System.out.println("Masukan Data mahasiswa ke-" + (i +
1));
    System.out.print("Nim: ");
    String nim = input.nextLine();
    System.out.print("Nama: ");
    String nama = input.nextLine();
    System.out.print("Kelas: ");
    String kelas = input.nextLine();
    System.out.print("IPK: ");
    double IPK = input.nextDouble();
    input.nextLine();

    Mahasiswa26 m = new Mahasiswa26(nim, nama, kelas,
IPK);
    list.tambah(m);
    System.out.println("-----");
}
```


Latihan

Kode program

```
package Jobsheet7;

public class Dosen26 {
    String kode;
    String nama;
    Boolean jenis;
    int usia;

    Dosen26() {

    }

    Dosen26(String kd, String name, Boolean jk, int age) {
        kode = kd;
        nama = name;
        jenis = jk;
        usia = age;
    }

    void tampil () {
        String jns = jenis ? "Laki-Laki" : "Perempuan" ;
        System.out.println("Kode: " + kode);
        System.out.println("Nama: " + nama);
        System.out.println("Jenis: " + jns);
        System.out.println("Usia: " + usia);
    }
}
```

```

package Jobsheet7;

public class DataDosen26 {
    Dosen26[] dataDosen = new Dosen26[10];
    int idx = 0;

    void tambah(Dosen26 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh!");
        }
    }

    void tampil() {
        for (Dosen26 dsn: dataDosen) {
            if (dsn != null) {
                String jns = dsn.jenis ? "Laki-Laki" : "Perempuan" ;
                System.out.println();
                System.out.println("Kode: " + dsn.kode);
                System.out.println("Nama: " + dsn.nama);
                System.out.println("Jenis: " + jns);
                System.out.println("Usia: " + dsn.usia);
                System.out.println("-----");
            }
        }
    }

    void tampilPosisi (int x, int pos) {
        if (pos!= -1) {
            System.out.println("data Dosen dengan Usia :" + x + "
ditemukan pada indeks " + pos);
        } else {
            System.out.println("data " + x + "tidak ditemukan");
        }
    }

    void tampilDataSearch (int x, int pos) {
        if (pos !=-1) {
            System.out.println("kode\t : " + dataDosen[pos].kode);
            System.out.println("nama\t : " + dataDosen[pos].nama);
            System.out.println("jenis\t : " + dataDosen[pos].jenis);
            System.out.println("usia\t : " + x);
        } else {
            System.out.println("Data Dosen dengan Usia " + x + "tidak
ditemukan");
        }
    }

    void tampilPosisiNama (String x, int pos) {
        if (pos!= -1) {
            System.out.println("data Dosen dengan Nama :" + x + "
ditemukan pada indeks " + pos);
        } else {
            System.out.println("data " + x + "tidak ditemukan");
        }
    }
}

```

```

void tampilDataSearchNama (String x, int pos) {
    if (pos != -1) {
        System.out.println("kode\t : " + dataDosen[pos].kode);
        System.out.println("nama\t : " + x);
        System.out.println("jenis\t : " + dataDosen[pos].jenis);
        System.out.println("usia\t : " + dataDosen[pos].usia);
    } else {
        System.out.println("Data Dosen dengan Nama " + x + "tidak
ditemukan");
    }
}

void SortingASC() {
    for (int i = 0; i < idx - 1; i++) {
        for (int j = 1; j < idx - i; j++) {
            if (dataDosen[j].usia < dataDosen[j - 1].usia) {
                Dosen26 tmp = dataDosen[j];
                dataDosen[j] = dataDosen[j - 1];
                dataDosen[j - 1] = tmp;
            }
        }
    }
}

void SortingDSC() {
    for (int i = 0; i < idx; i++) {
        Dosen26 temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
}

int sequentialSearching(String cari) {
    int posisi = -1;
    int count = 0;

    for (int j = 0; j < dataDosen.length; j++) {
        if (dataDosen[j] != null &&
dataDosen[j].nama.equalsIgnoreCase(cari)) {
            if (posisi == -1) {
                posisi = j;
            }
            count++;
        }
    }

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

    return posisi;
}

```

```

int BinarySearch(int caril, int left, int right) {
    int mid;
    int count = 0;
    int posisi = -1;

    if (right >= left) {
        mid = (left + right) / 2;

        if (dataDosen[mid].usia == caril) {
            posisi = mid;
            count++;
            int l = mid - 1;
            while (l >= left && dataDosen[l].usia == caril) {
                count++;
                l--;
            }
            int r = mid + 1;
            while (r <= right && dataDosen[r].usia == caril) {
                count++;
                r++;
            }
        } else if (dataDosen[mid].usia > caril) {
            return BinarySearch(caril, left, mid - 1);
        } else {
            return BinarySearch(caril, mid + 1, right);
        }
    }

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

    return posisi;
}

```

```

package Jobsheet7;

import java.util.Scanner;

import Jobsheet6.DataDosen;

public class DosenMain26 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        DataDosen26 dataDosen = new DataDosen26();

        dataDosen.tambah(new Dosen26("D001", "Andi", true, 35));
        dataDosen.tambah(new Dosen26("D002", "Ani", false, 40));
        dataDosen.tambah(new Dosen26("D003", "Budi", true, 40));
        dataDosen.tambah(new Dosen26("D004", "Ani", false, 45));

        System.out.println("-----
--");
        System.out.println("Pencarian data");
        System.out.println("-----
--");
        System.out.println("masukan nama dosen yang dicari: ");
        System.out.print("nama: ");
        String cari = input.nextLine();

        System.out.println("menggunakan sequential searching");
        int posisi = dataDosen.sequentialSearching(cari);
        if (posisi == -1) {
            System.out.println("Data dosen dengan nama " + cari + "
tidak ditemukan.");
        } else {
            dataDosen.tampilPosisiNama(cari, posisi);
            dataDosen.tampilDataSearchNama(cari, posisi);
        }

        System.out.println("-----
");
        System.out.println("Pencarian data");
        System.out.println("-----
");
        System.out.println("masukan usia dosen yang dicari: ");
        System.out.print("usia: ");
        int caril = input.nextInt();
        System.out.println("-----");
        System.out.println("menggunakan binary search");
        System.out.println("-----");
        int posisi2 = dataDosen.BinarySearch(caril, 0, dataDosen.idx-1);
        if (posisi2 == -1) {
            System.out.println("Data dosen dengan usia " + caril + "
tidak ditemukan.");
        } else {
            dataDosen.tampilPosisi(caril, posisi2);
            dataDosen.tampilDataSearch(caril, posisi2);
        }
    }
}

```

```

        System.out.println("Data Dosen sebelum disorting:");
        dataDosen.tampil();

        System.out.println("Data telah diurutkan secara ASC berdasarkan
usia:");
        dataDosen.SortingASC();
        dataDosen.tampil();

        System.out.println("Data telah diurutkan secara DSC berdasarkan
usia:");
        dataDosen.SortingDSC();
        dataDosen.tampil();
    }
}

```

Output

```

PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD\Jobsheet7>
ionMessages' '-cp' 'C:\Users\THINKPAD\AppData\Roaming\Code\User\worksp
roject\bin' 'Jobsheet7.DosenMain26'
-----
Pencarian data
-----
masukan nama dosen yang dicari:
nama: ani
menggunakan sequential searching
Peringatan: Ditemukan lebih dari satu dosen dengan nama yang sama!
data Dosen dengan Nama :ani ditemukan pada indeks 1
kode      : D002
nama      : ani
jenis     : false
usia      : 40
-----
Pencarian data
-----
masukan usia dosen yang dicari:
usia: 40
-----
menggunakan binary search
-----
Peringatan: Ditemukan lebih dari satu dosen dengan usia yang sama!
data Dosen dengan Usia :40 ditemukan pada indeks 1
kode      : D002
nama      : Ani
jenis     : false
usia      : 40

```

Data Dosen sebelum disorting:

Kode: D001
Nama: Andi
Jenis: Laki-Laki
Usia: 35

Kode: D002
Nama: Ani
Jenis: Perempuan
Usia: 40

Kode: D003
Nama: Budi
Jenis: Laki-Laki
Usia: 40

Kode: D004
Nama: Ani
Jenis: Perempuan
Usia: 45

Data telah diurutkan secara ASC berdasarkan usia:

Kode: D001
Nama: Andi
Jenis: Laki-Laki
Usia: 35

Kode: D002
Nama: Ani
Jenis: Perempuan
Usia: 40

```
Kode: D003
Nama: Budi
Jenis: Laki-Laki
Usia: 40
-----

Kode: D004
Nama: Ani
Jenis: Perempuan
Usia: 45
-----
Data telah diurutkan secara DSC berdasarkan usia:

Kode: D004
Nama: Ani
Jenis: Perempuan
Usia: 45
-----

Kode: D002
Nama: Ani
Jenis: Perempuan
Usia: 40
-----

Kode: D003
Nama: Budi
Jenis: Laki-Laki
Usia: 40
-----

Kode: D001
Nama: Andi
Jenis: Laki-Laki
Usia: 35
-----
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD\Jobsheet7>
```