

# **LAPORAN PRAKTIKUM**

## **JOBSHEET 7**



**VANESA MARDIANA PUTRI**

**244107020129 / 23**

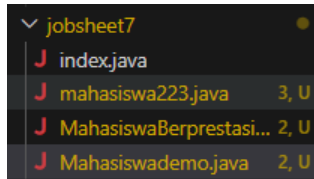
**KELAS TI 1B**

**PRODI D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

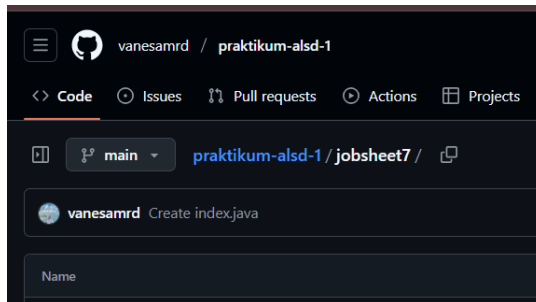
## 1. Percobaan 1

Langkah – Langkah :

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

```
void tampilPoisisi(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<no presensi>, 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);
    }
}
```

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

```
int jumMhs = 5;

for (int i = 0; i < jumMhs; i++) {
    System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));
    System.out.print(s:"NIM: ");
    String nim = sc.nextLine();
    System.out.print(s:"Nama: ");
    String nama = sc.nextLine();
    System.out.print(s:"Kelas: ");
    String kelas = sc.nextLine();
    System.out.print(s:"IPK: ");
    String ip = sc.nextLine();
    double ipk = Double.parseDouble(ip);
    System.out.println(x:"-----");
    list.tambah (new mahasiswa223(nim, nama,kelas,ipk));
}

list.tampil();

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 = sc.nextDouble();

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

7. Run kode.

```

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
-----

```

```

NIM: 111
Nama: adi
Kelas: 2
IPK: 3.1
-----
NIM: 222
Nama: ila
Kelas: 2
IPK: 3.2
-----
NIM: 333
Nama: lia
Kelas: 2
IPK: 3.3
-----
NIM: 444
Nama: susi
Kelas: 2
IPK: 3.4
-----
NIM: 555
Nama: anita
Kelas: 2
IPK: 3.5
-----
Pencarian data

Masukkan ipk mahasiswa yang dicari:
IPK:
3.5
menggunakan sequential searching
data mahasiswa dengan IPK : 3.5 ditemukan pada indeks 4
nim      : 555
nama     : anita
kelas    : 2
ipk      : 3.5
PS.D:\kuliab\sm2\alisd\praktikum\alisd-1\

```

Pertanyaan jawaban :

- Perbedaan antara dua metode (tampilPosisi dan tampilDataSearch) adalah : jika tampilPosisi akan menampilkan posisi (letak/indeks) dari data mahasiswa berdasarkan IPK x. Sementara tampilDataSearch akan menampilkan data lengkap mahasiswa yang memiliki IPK x (jika ditemukan).
- Fungsi break adalah untuk menghentikan perulangan saat data mahasiswa dengan IPK yang dicari sudah ditemukan, agar tidak lanjut mencari data yang selanjutnya.

## 2. Percobaan 2

Langkah – Langkah :

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.

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

3. Run kode.

```

Masukkan Data Mahasiswa ke-1
NIM: 111
Nama: adi
Kelas: 2
IPK: 3.1
-----
Masukkan Data Mahasiswa ke-2
NIM: 222
Nama: lia
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
-----

```

```

-----
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
PS D:\kuliah\smt 2\alsd\praktikum-alsd-1>

```

Pertanyaan jawaban :

```

- 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);
    }
}
-

```

- Jika IPK tidak terurut, maka program akan tetap berjalan atau tidak error tapi hasil pencarian bisa salah atau tidak ditemukan, meskipun data sebenarnya ada. Hal ini dikarenakan binary search akan berjalan atau bekerja jika data sudah urut.

```

- 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) { This block of commented-out lines
        // return findBinarySearch(cari, left, mid - 1);
        } else if (listMhs[mid].ipk < cari) { //modifikasi no.4
            return findBinarySearch(cari, left, mid - 1); //modifikasi no.4
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}
-

```

```

MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23();
//int jumMhs=5; This block of commented-out code is never used.
int jumMhs = sc.nextInt();
sc.nextLine();

```

### 3. Tugas

1.

```

void PencarianDataSequential23() {
    Scanner input = new Scanner(System.in);
    System.out.println("Masukkan nama dosen yang ingin dicari: ");
    String nama = input.nextLine();

    boolean found = false;
    int count = 0;

    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equals(nama)) {
            found = true;
            count++;

            System.out.println("Dosen ditemukan!");
            System.out.println("Kode: " + dataDosen[i].kode);
            System.out.println("Nama: " + dataDosen[i].nama);
            System.out.println("Jenis Kelamin: " + dataDosen[i].jenisKelamin);
            System.out.println("Usia: " + dataDosen[i].usia);
            System.out.println("-----");
        }
    }

    if (!found) {
        System.out.println("Dosen dengan nama " + nama + " tidak ditemukan.");
    } else if (count > 1) {
        System.out.println("Peringatan! Ditemukan " + count + " dosen dengan nama " + nama + ".");
    }
}

```

2.

```

void PencarianDataBinary23() {
    Scanner input = new Scanner(System.in);
    System.out.println("Masukkan nama dosen yang ingin dicari: ");
    String nama = input.nextLine();

    sortingDC();

    int left = 0;
    int right = idx - 1;
    boolean found = false;
    int count = 0;

    while (left <= right) {
        int mid = (left + right) / 2;
        int compare = dataDosen[mid].nama.compareTo(nama);

        if (compare == 0) {
            found = true;
            count++;

            System.out.println("Dosen ditemukan!");
            System.out.println("Kode: " + dataDosen[mid].kode);
            System.out.println("Nama: " + dataDosen[mid].nama);
            System.out.println("Jenis Kelamin: " + dataDosen[mid].jenisKelamin);
            System.out.println("Usia: " + dataDosen[mid].usia);
            System.out.println("-----");
        }
    }
}

```

3.

```

int temp = mid - 1;
while (temp >= left && dataDosen[temp].nama.equals(nama)) {
    count++;
    System.out.println("Dosen ditemukan!");
    System.out.println("Kode: " + dataDosen[temp].kode);
    System.out.println("Nama: " + dataDosen[temp].nama);
    System.out.println("Jenis Kelamin: " + dataDosen[temp].jenisKelamin);
    System.out.println("Usia: " + dataDosen[temp].usia);
    System.out.println("-----");
    temp--;
}

temp = mid + 1;
while (temp <= right && dataDosen[temp].nama.equals(nama)) {
    count++;
    System.out.println("Dosen ditemukan!");
    System.out.println("Kode: " + dataDosen[temp].kode);
    System.out.println("Nama: " + dataDosen[temp].nama);
    System.out.println("Jenis Kelamin: " + dataDosen[temp].jenisKelamin);
    System.out.println("Usia: " + dataDosen[temp].usia);
    System.out.println("-----");
    temp++;
}

break;
} else if (compare < 0) {
    left = mid + 1;
} else {
    right = mid - 1;
}
}

if (!found) {
    System.out.println("Dosen dengan nama " + nama + " tidak ditemukan.");
} else if (count > 1) {
    System.out.println("Peringatan! Ditemukan " + count + " dosen dengan nama " + nama + ".");
}
}

```