

Nama : Siti Nikmatus Sholihah

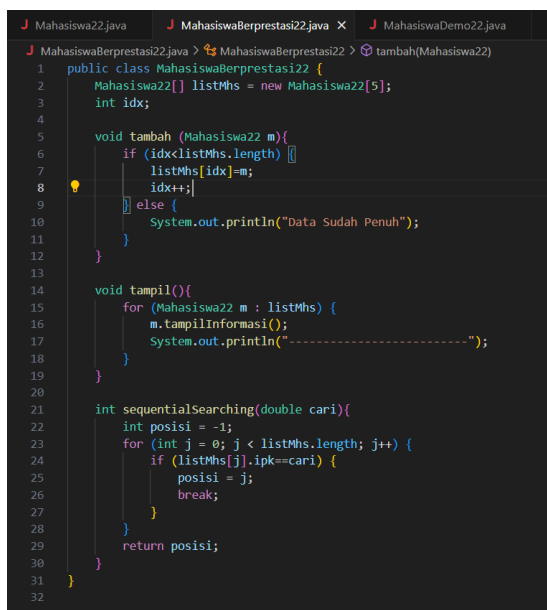
NIM : 244107020014

Kelas : TI 1B

## Jobsheet 7 – Searching

### 1. Searching / Pencarian Menggunakan Algoritma Sequential Search

- 1.1 Pada pertemuan Jobsheet 7 ini akan menggunakan class Mahasiswa, MahasiswaBerprestasi, dan MahasiswaDemo pada pertemuan Jobsheet 6 sebelumnya.
- 1.2 Buat folder baru bernama Jobsheet7 di dalam repository Praktikum ASD, kemudian buka ketiga class dari Jobsheet 6 tersebut dan copy ke folder Jobsheet 7.
- 1.3 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.



```
J Mahasiswa22.java  J MahasiswaBerprestasi22.java  J MahasiswaDemo22.java
J MahasiswaBerprestasi22.java > MahasiswaBerprestasi22 > tambah(Mahasiswa22)
1 public class MahasiswaBerprestasi22 {
2     Mahasiswa22[] listMhs = new Mahasiswa22[5];
3     int idx;
4
5     void tambah (Mahasiswa22 m){
6         if (idx<listMhs.length) {
7             listMhs[idx]=m;
8             idx++;
9         } else {
10            System.out.println("Data Sudah Penuh");
11        }
12    }
13
14    void tampil(){
15        for (Mahasiswa22 m : listMhs) {
16            m.tampilInformasi();
17            System.out.println("-----");
18        }
19    }
20
21    int sequentialSearching(double cari){
22        int posisi = -1;
23        for (int j = 0; j < listMhs.length; j++) {
24            if (listMhs[j].ipk==cari) {
25                posisi = j;
26                break;
27            }
28        }
29        return posisi;
30    }
31 }
32
33
```

- 1.4 Buatlah method tampilPoisisi bertipe void dan Deklarasikan isi dari method tampilPoisisi pada class MahasiswaBerprestasi.

```
J Mahasiswa22.java J MahasiswaBerprestasi22.java X J MahasiswaDemo22.java 1
J MahasiswaBerprestasi22.java > J MahasiswaBerprestasi22 > tampil()
1 public class MahasiswaBerprestasi22 {
2     Mahasiswa22[] listMhs = new Mahasiswa22[5];
3     int idx;
4
5     void tambah (Mahasiswa22 m){
6         if (idx<listMhs.length) {
7             listMhs[idx]=m;
8             idx++;
9         } else {
10             System.out.println(x:"Data Sudah Penuh");
11         }
12     }
13
14     void tampil(){
15         for (Mahasiswa22 m : listMhs) {
16             m.tampilInformasi();
17             System.out.println(x:"-----");
18         }
19     }
20
21     void tampilPosisi(double x, int pos){
22         if (pos!=-1) {
23             System.out.println("data Mahasiswa dengan IPK: " +x+ " ditemukan pada index " + pos);
24         } else{
25             System.out.println("data " +x+ " tidak ditemukan");
26         }
27     }
28 }
```

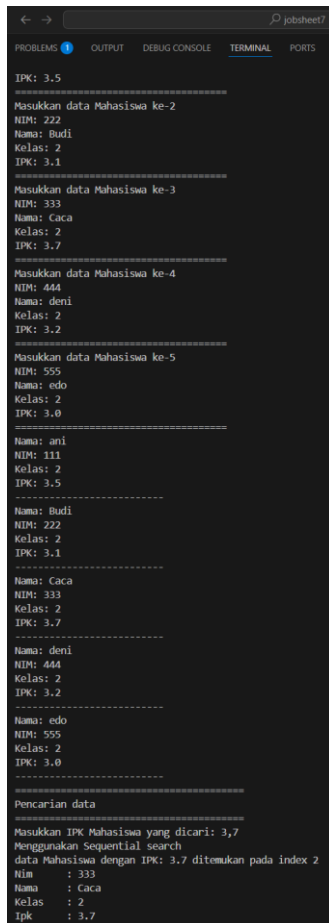
- 1.5 Pada class MahasiswaBerprestasi, buatlah method tampilDataSearch bertipe void dan Deklarasikan isi dari method tampilDataSearch .

```
J MahasiswaBerprestasi22.java > J MahasiswaBerprestasi22 > tampilDataSearch(double, int)
1 public class MahasiswaBerprestasi22 {
2
3     void tampil(){
4         for (Mahasiswa22 m : listMhs) {
5             m.tampilInformasi();
6             System.out.println(x:"-----");
7         }
8     }
9
10    void tampilPosisi(double x, int pos){
11        if (pos!=-1) {
12            System.out.println("data Mahasiswa dengan IPK: " +x+ " ditemukan pada index " + pos);
13        } else{
14            System.out.println("data " +x+ " tidak ditemukan");
15        }
16    }
17
18    void tampilDataSearch(double x, int pos){
19        if (pos!=-1) {
20            System.out.println("Nama: " +listMhs[pos].nama);
21            System.out.println("Nim: " +listMhs[pos].nim);
22            System.out.println("Kelas: " +listMhs[pos].kelas);
23            System.out.println("IPK: " +listMhs[pos].ipk);
24        } else {
25            System.out.println("data Mahasiswa dengan ipk " +x+ " tidak ditemukan");
26        }
27    }
28 }
```

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

```
J Mahasiswa22.java J MahasiswaBerprestasi22.java X J MahasiswaDemo22.java X
J MahasiswaDemo22.java > J MahasiswaDemo22 > main(String[])
1 public class MahasiswaDemo22 {
2     public static void main(String[] args) {
3         MahasiswaBerprestasi22 list = new MahasiswaBerprestasi22();
4         Scanner sc = new Scanner(System.in);
5
6         System.out.print(s:"Masukkan jumlah mahasiswa: ");
7         int jumMhs = sc.nextInt();
8         sc.nextLine();
9
10        for (int i = 0; i < jumMhs; i++) {
11            System.out.println("Masukkan data Mahasiswa ke-" + (i + 1));
12            System.out.print(s:"NIM: ");
13            String nim = sc.nextLine();
14            System.out.print(s:"Nama: ");
15            String nama = sc.nextLine();
16            System.out.print(s:"Kelas: ");
17            String kelas = sc.nextLine();
18            System.out.print(s:"IPK: ");
19            double ipk = Double.parseDouble(sc.nextLine());
20            System.out.println(x:"-----");
21            list.tambah(new Mahasiswa22(nim, nama, kelas, ipk));
22        }
23
24        //melakukan pencarian menggunakan Sequential Search
25        list.tampil();
26        System.out.println(x:"-----");
27        System.out.println(x:"Pencarian data");
28        System.out.println(x:"-----");
29        System.out.print(s:"Masukkan IPK Mahasiswa yang dicari: ");
30        double cari = sc.nextDouble();
31        System.out.println(x:"menggunakan sequential search");
32        double posisi = list.sequentialSearching(cari);
33        int pss = (int)posisi;
34        list.tampilPosisi(cari, pss);
35        list.tampilDataSearch(cari, pss);
36    }
37 }
```

## 1.7 Jalankan dan amati hasilnya.



```
IPK: 3.5
=====
Masukkan data Mahasiswa ke-2
NIM: 222
Nama: Budi
Kelas: 2
IPK: 3.1
=====
Masukkan data Mahasiswa ke-3
NIM: 333
Nama: Caca
Kelas: 2
IPK: 3.7
=====
Masukkan data Mahasiswa ke-4
NIM: 444
Nama: deni
Kelas: 2
IPK: 3.2
=====
Masukkan data Mahasiswa ke-5
NIM: 555
Nama: edo
Kelas: 2
IPK: 3.0
=====
Nama: ani
NIM: 111
Kelas: 2
IPK: 3.5
=====
Nama: Budi
NIM: 222
Kelas: 2
IPK: 3.1
=====
Nama: Caca
NIM: 333
Kelas: 2
IPK: 3.7
=====
Nama: deni
NIM: 444
Kelas: 2
IPK: 3.2
=====
Nama: edo
NIM: 555
Kelas: 2
IPK: 3.0
=====
Pencarian data
=====
Masukkan IPK Mahasiswa yang dicari: 3.7
Menggunakan Sequential search
data Mahasiswa dengan IPK: 3.7 ditemukan pada index 2
Nim      : 333
Nama     : Caca
Kelas   : 2
IpK      : 3.7
```

### Pertanyaan

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

Jawab:

- a. **tampilPosisi**: Menampilkan posisi indeks dari data yang dicari (jika ditemukan) atau pesan tidak ditemukan(jika tidak ditemukan). Contoh output:  
"data mahasiswa dengan IPK :3.5 ditemukan pada indeks 3".
- b. **tampilDataSearch**: Menampilkan detail lengkap data mahasiswa (NIM, nama, kelas, IPK) jika ditemukan, atau pesan error jika tidak ditemukan. Contoh output:  
"nim: 444, nama: ila, kelas: 2, ipk: 3.5".

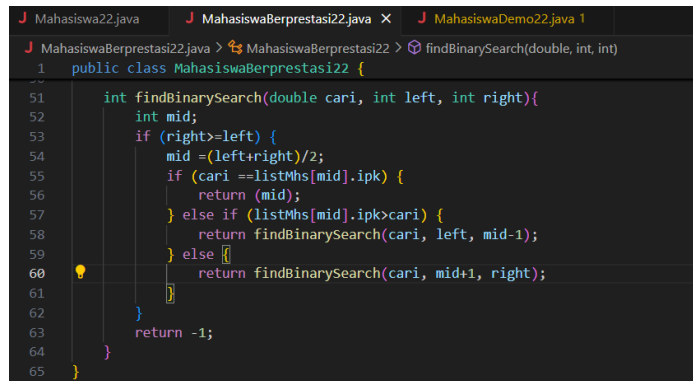
2. Jelaskan fungsi break pada kode program dibawah ini!

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

Jawab: break digunakan untuk menghentikan perulangan (for) segera setelah data yang dicari ditemukan.

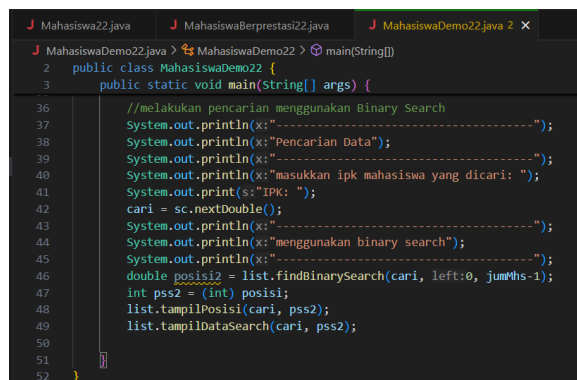
### 3. Searching / Pencarian Menggunakan Algoritma Binary Search

- 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.



```
1 public class MahasiswaBerprestasi22 {  
2  
51     int findBinarySearch(double cari, int left, int right){  
52         int mid;  
53         if (right >= left) {  
54             mid = (left+right)/2;  
55             if (cari == listMhs[mid].ipk) {  
56                 return (mid);  
57             } else if (listMhs[mid].ipk > cari) {  
58                 return findBinarySearch(cari, left, mid-1);  
59             } else {  
60                 return findBinarySearch(cari, mid+1, right);  
61             }  
62         }  
63         return -1;  
64     }  
65 }
```

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



```
1 public class MahasiswaDemo22 {  
2     public static void main(String[] args) {  
3  
36         //melakukan pencarian menggunakan Binary Search  
37         System.out.println(x: "-----");  
38         System.out.println(x: "Pencarian Data");  
39         System.out.println(x: "-----");  
40         System.out.println(x: "masukkan ipk mahasiswa yang dicari: ");  
41         System.out.print(s: "IPK: ");  
42         cari = sc.nextDouble();  
43         System.out.println(x: "-----");  
44         System.out.println(x: "menggunakan binary search");  
45         System.out.println(x: "-----");  
46         double posisi2 = list.findBinarySearch(cari, left:0, jumMhs-1);  
47         int pss2 = (int) posisi;  
48         list.tampilPosisi(cari, pss2);  
49         list.tampilDataSearch(cari, pss2);  
50  
51     }  
52 }
```

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

```
← → jobsheet7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\Praktikum-ASD\jobsheet7>cd "d:\Praktikum-ASD\jobsheet7"
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -D...
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -D...
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: lila
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: lila
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 search
data Mahasiswa dengan IPK: 3.7 ditemukan pada index 4
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 index 4
Nim      : 555
Nama     : anita
Kelas   : 2
IpK      : 3.7
```

## Pertanyaan

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

Jawab:

Proses divide terjadi pada kode program: `mid =(left+right)/2;`

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

Jawab:

Proses conquer terjadi pada saat membandingkan nilai cari dengan `listMhs[mid].ipk` dan memutuskan untuk mencari dibagian kiri atau kanan. Kode program:

```
if (cari == listMhs[mid].ipk) {
    return mid; // Data ditemukan
} else if (listMhs[mid].ipk > cari) {
```

```

return findBinarySearch(cari, left, mid - 1); // Cari di kiri
} else {
return findBinarySearch(cari, mid + 1, right); // Cari di kanan
}

```

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

Jawab:

**Tidak bisa berjalan**, karena binary search memerlukan data terurut (ASC/DESC) untuk membagi area pencarian secara efektif. Jika data acak, hasil pencarian mungkin salah atau tidak ditemukan.

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 .

Jawab:

**Hasil tidak sesuai**, karena kode default binary search hanya bekerja untuk data terurut ASC. Agar hasil bisa sesuai dengan memasukkan ipk terbesar ke terkecil maka kode yang harus diubah adalah sebagai berikut:

```

else if (listMhs[mid].ipk < cari) { // Ubah '>' menjadi '<' untuk DESC
return findBinarySearch(cari, left, mid - 1);
}

```

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

Jawab:

Ganti jumMhs = 5 dengan input dinamis

```

System.out.print("Masukkan jumlah mahasiswa: ");
int jumMhs = sc.nextInt();
sc.nextLine(); // Membersihkan newline

```