

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET 7



SHAFIQA NABILA MAHARANI KHOIRUNNISA

244107020221

TI -1B

7.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

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

```
-----  
NIM: 111  
Nama: adi  
Kelas: 2  
IPK: 3.6  
-----
```

```
NIM: 222  
Nama: tio  
Kelas: 2  
IPK: 3.8  
-----
```

```
NIM: 333  
Nama: ila  
Kelas: 2  
IPK: 3.0  
-----
```

```
NIM: 444  
Nama: lia  
Kelas: 2  
IPK: 3.5  
-----
```

```
-----  
NIM: 555  
Nama: fia  
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 C:\Users\fika\MATKUL SEMESTER 2\praktikum-ASD\Jobsheet 7> █
```

JAWABAN PERTANYAAN :

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

- tampilPosisi :

-Metode ini hanya menampilkan posisi indeks dari mahasiswa yang memiliki IPK tertentu.

-Jika data ditemukan (posisi tidak -1), maka akan mencetak informasi bahwa data mahasiswa dengan IPK tertentu ditemukan pada indeks yang sesuai, dan jika tidak ditemukan, akan mencetak pesan bahwa data dengan IPK tersebut tidak ditemukan.

- tampilDataSearch :

-Metode ini lebih rinci, karena menampilkan data lengkap mahasiswa yang memiliki IPK tertentu.

-Jika data ditemukan, akan mencetak NIM, Nama, Kelas, dan IPK mahasiswa tersebut, dan jika tidak ditemukan, akan mencetak pesan bahwa data dengan IPK tersebut tidak ditemukan.

2. Jelaskan fungsi break pada kode program dibawah ini!

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

- sebuah loop (seperti for atau while) yang digunakan untuk mencari mahasiswa dengan IPK tertentu (cari), Saat menemukan mahasiswa pertama yang sesuai (listMhs[j].ipk == cari) variabel posisi diset ke nilai j (indeks mahasiswa yang ditemukan).
- Perintah break; menghentikan perulangan agar tidak perlu mencari lagi, tanpa break perulangan akan terus berjalan meskipun data sudah ditemukan, yang dapat membuang waktu dan sumber daya.

7.3. Searching / Pencarian Menggunakan Binary Search

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

```
-----  
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 C:\Users\fika\MATKUL SEMESTER 2\praktikum-ASD\Jobsheet 7> |
```

JAWABAN PERTANYAAN :

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

- $mid = (left + right) / 2;$

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

- ```
else if (listMhs[mid].ipk > cari) {
 return findBinarySearch(cari, left, mid - 1);
} else {
 return findBinarySearch(cari, mid + 1, right);
}
```

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

- Jika data IPK tidak terurut, program binary search tidak akan dapat menemukan nilai yang dicari dengan benar, dan hasilnya bisa salah atau tidak ditemukan sama sekali.

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.

- Tidak sesuai
- Modifikasi

```
// MODIFIKASI PERTANYAAN NO.4
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;
}
```

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

```
System.out.println(x:"Masukkan jumlah mahasiswa: "); Replace this use of System.out by a logger.
int jumMhs = sc.nextInt();
sc.nextLine();

MahasiswaBerprestasi20 list = new MahasiswaBerprestasi20(jumMhs);
```

```
Masukkan jumlah mahasiswa:
3
Masukkan Data Mahasiswa ke-1
NIM: 111
Nama: fika
Kelas: 1
IPK: 3.5

Masukkan Data Mahasiswa ke-2
NIM: 222
Nama: abid
Kelas: 1
IPK: 3.9

Masukkan Data Mahasiswa ke-3
NIM: 333
Nama: kanari
Kelas: 1
IPK: 3.4

Pencarian data

Masukkan ipk mahasiswa yang dicari:
IPK:
3.9
menggunakan binary search

data mahasiswa dengan IPK :3.9 ditemukan pada indeks 1
nim : 222
nama : abid
kelas : 1
ipk : 3.9
```



## LATIHAN PRAKTIKUM :

A. ...

```
void PencarianDataSequential20() { Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
 Scanner input = new Scanner();
 System.out.println(x: "Masukkan nama dosen yang ingin dicari: "); Replace this use of System.out by a logger.
 String nama = input.nextLine();
 for (int i; i < idx; i++){
 if (dataDosen[i].nama == nama){ Strings and Boxed types should be compared using "equals()".
 System.out.println(x: "Dosen ditemukan!"); Define a constant instead of duplicating this literal "Dosen di
 System.out.println("Kode: " + dataDosen[i].kode); Define a constant instead of duplicating this literal "
 System.out.println("Nama: " + dataDosen[i].nama); Define a constant instead of duplicating this literal "
 System.out.println("Jenis Kelamin: " + dataDosen[i].jenisKelamin); Define a constant instead of duplicati
 System.out.println("Usia: " + dataDosen[i].usia); Define a constant instead of duplicating this literal "
 }
 }
}
```

B. ...

```
void PencarianDataBinary20(){ Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
 Scanner input = new Scanner();
 System.out.println(x: "Masukkan nama dosen yang ingin dicari: ");
 String nama = input.nextLine();
 int mid = (idx - 1) / 2;
 int left = 0;
 int right = idx - 1;
 while (left <= right) {
 if (dataDosen[mid].nama == nama){
 count++;
 System.out.println(x: "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);
 }
 }
}
```

C. ...

```
void PencarianDataSequential20() { Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
 Scanner input = new Scanner();
 System.out.println(x: "Masukkan nama dosen yang ingin dicari: "); Replace this use of System.out by a logger.
 String nama = input.nextLine();
 for (int i; i < idx; i++){
 if (dataDosen[i].nama == nama) { Define a constant instead of duplicating this literal "Nama: " 4 times. [+4
 System.out.println("Nama: " + dataDosen[i].nama); locations] sonarqube(java:S1192)
 System.out.println("Jenis Kelamin: " + dataDosen[i].jenisKelamin); Define a constant instead of duplicating this literal "Na
 System.out.println("Usia: " + dataDosen[i].usia); Define a constant instead of duplicating this literal "Us
 }
 }

 if (!found) {
 System.out.println("Dosen dengan nama " + nama + " tidak ditemukan."); Replace this use of System.out by a logg
 } else if (count > 1) {
 System.out.println("Peringatan! Ditemukan " + count + " dosen dengan nama " + nama + "."); Replace this use of
 }
}
```

```

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

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

```