

**LAPORAN TUGAS ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET 7**



NAMA : RADITYA RIEFKI  
KELAS : TI 1E  
ABSEN : 23

## JOBSHEET VII SEARCHING

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

#### 1.Kode Program

##### Mahasiswa Berprestasi

```
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;
}

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");
    }
}

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");
    }
}
```

##### Mahasiswa Demo

```
package jobsheet7;

import java.util.Scanner;

public class MahasiswaDemo23 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int jumMhs = 5;

        MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23(jumMhs);

        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: ");
        double ipk = sc.nextDouble();
        sc.nextLine();

        Mahasiswa23 m = new Mahasiswa23(nim, nama, kelas, ipk);
        list.tambah(m);
        System.out.println();
    }

    System.out.println("-----");
    System.out.println("Pencarian Data");
    System.out.println("-----");
    System.out.print("Masukkan IPK Mahasiswa Yang Ingin 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);
}
}

```

## 2. Output

```

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: 3.5

-----
Pencarian Data
-----
Masukkan IPK Mahasiswa Yang Ingin 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:\CollegeFile\SM1 2\ALSD>

```

### 7.2.3. Pertanyaan

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

- method tampilPosisi digunakan untuk memberi tahu posisi indeks ke berapa dalam data array

Method tampilDataSearch digunakan untuk menampilkan informasi lengkap tentang mahasiswa yang dicari berdasarkan IPK

2. Jelaskan fungsi break pada kode program dibawah ini!

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

- Fungsi break pada kode diatas digunakan jika kondisi listMhs[j].ipk sama dengan input cari maka akan berhenti

### 7.3. Searching / Pencarian Menggunakan Binary Search

1. Kode Program

Mahasiswa Berprestasi

```
int findBinarySearch(double cari, int left, int right){  
    int mid;  
    if (right >= left) {  
        mid = left + (right - left) / 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;  
}
```

## Mahasiswa Demo

```
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();
    sc.nextLine();
    System.out.println("-----");
    System.out.println("Menggunakan Binary Search");
    System.out.println("-----");
    double posisi2 = list.findBinarySearch(cari, 0, jumMhs-1);
    int pss2 = (int) posisi2;
    list.tampilPosisi(cari, pss2);
    list.tampilDataSearch(cari, pss2);
```

## 2. Output

```
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 D:\CollegeFile\SMT 2\ALSD>
```

### 7.3.3. Pertanyaan

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

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

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

- `return findBinarySearch(cari, left, mid - 1);`

Dan

`return findBinarySearch(cari, mid + 1, right);`

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

- binary search hanya berfungsi jika data yang ingin dicari sudah urut jika data belum urut maka harus diurutkan terlebih dahulu

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

```
int findBinarySearch(double cari, int left, int right){
    int mid;
    if (right >= left) {
        mid = left + (right - left) / 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;
}
```

Mengubah kondisi dari `(listMhs[mid].ipk > cari)` menjadi `(listMhs[mid].ipk < cari)`

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

### Penambahan kode pada program class MahasiswaDemo23

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
  
    System.out.println(x:"Masukkan Jumlah Mahasiswa: ");  
    int jumMhs = 0;  
    jumMhs = sc.nextInt();  
  
    MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23(jumMhs);
```

### Penambahan kode program pada class MahasiswaBerprestasi23

```
public class MahasiswaBerprestasi23 {  
    Mahasiswa23[] listMhs;  
    int idx;  
  
    public MahasiswaBerprestasi23(int jumlah){  
        listMhs = new Mahasiswa23[jumlah];  
        idx = 0;
```

```
    void tampil() {  
        for(Mahasiswa23 m : listMhs){  
            if (m != null) {  
                m.tampilInformasi();  
                System.out.println(x:"-----");  
            }  
        }  
    }  
}
```

### OUTPUT

```
Masukkan Jumlah Mahasiswa: 3  
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
```

## 7.5. Latihan Praktikum

1. Pada Latihan praktikum pertemuan sebelumnya pada Jobsheet 6 yang terdapat 3 class yaitu Dosen, DataDosen, dan DosenDemo, tambahkan method:

a. PencarianDataSequential : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.

b. PencarianDataBinary : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.

c. Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

### DataDosen23

```
void pencarianDataSequential(String cariNama) {
    int jumlahHasil = 0;
    for (int i = 0; i < arrayDosen23.length; i++) {
        if (arrayDosen23[i] != null && arrayDosen23[i].nama.equalsIgnoreCase(cariNama)) {
            System.out.println("Data dosen dengan nama : " + arrayDosen23[i].nama + " ditemukan pada indeks ke- " + i);
            System.out.println("Kode : " + arrayDosen23[i].kode);
            System.out.println("Nama : " + arrayDosen23[i].nama);
            if (arrayDosen23[i].jenisKelamin == true) {
                System.out.println(x:"Jenis Kelamin : Pria");
            } else {
                System.out.println(x:"Jenis Kelamin : Wanita");
            }
            System.out.println("Usia : " + arrayDosen23[i].usia);
            System.out.println();
            jumlahHasil++;
        }
    }
    if (jumlahHasil == 0) {
        System.out.println(x:"Data tidak ditemukan.");
    } else {
        System.out.println(x:"-----");
        System.out.println("PERINGATAN : Data ditemukan sebanyak " + jumlahHasil);
        System.out.println(x:"-----");
        if (jumlahHasil > 1) {
            System.out.println(x:"PERINGATAN : Terdapat data dosen berjumlah lebih dari 1.");
        }
    }
}

void pencarianDataBinary(int cariUsia){
```

```
void pencarianDataBinary(int cariUsia){
    int awal = 0, akhir = idx -1, tengah = -1;
    boolean ditemukan = false;

    while (awal <= akhir) {
        tengah = (awal + akhir) / 2;
        if (arrayDosen23[tengah].usia == cariUsia) {
            ditemukan = true;
            break;
        }
        if (arrayDosen23[tengah].usia < cariUsia) {
            awal = tengah + 1;
        } else {
            akhir = tengah - 1;
        }
    }

    if (!ditemukan) {
        System.out.println("Dosen dengan usia " + cariUsia + " tidak ditemukan");
        return;
    }

    int kiri = tengah, kanan = tengah;
    while (kiri >= 0 && arrayDosen23[kiri].usia == cariUsia) kiri--;
    while (kanan < idx && arrayDosen23[kanan].usia == cariUsia) kanan++;
    System.out.println("hasil pencarian dosen dengan usia " + cariUsia+": ");

    for (int i = kiri + 1; i < kanan; i++) {
        arrayDosen23[i].tampil();
    }
}
```



```

    if (kanan - kiri - 1 > 1) {
        System.out.println(x:"Peringatan! ada lebih dari 1 dosen dengan usia yang sama");
    }
}
}

```

## DosenMain

```

case 5:
    System.out.println(x:"Sequential Search");
    System.out.print(s:"Masukkan nama dosen yang dicari : ");
    String cari = sc.nextLine();
    dosen.sortingAsc();
    dosen.pencarianDataSequential(cari);
    break;
case 6:
    System.out.println(x:"Binary Search");
    System.out.print(s:"Masukkan usia dosen yang ingin dicari : ");
    int search = sc.nextInt();
    sc.nextLine();
    dosen.sortingAsc();
    dosen.pencarianDataBinary(search);
    break;

```

## OUTPUT

```

Pilih menu: 5
Sequential Search
Masukkan nama dosen yang dicari : adi
Data dosen dengan nama : adi ditemukan pada indeks ke- 0
Kode : 111
Nama : adi
Jenis Kelamin : Pria
Usia : 22

-----
PERINGATAN : Data ditemukan sebanyak 1
-----

```

```

Binary Search
Masukkan usia dosen yang ingin dicari : 23
hasil pencarian dosen dengan usia 23:
Data Dosen ke- 1
Kode: 22
Nama: oli
Jenis Kelamin: Wanita
Usia: 23

```