

**LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR
SEARCHING (SEQUENTIAL DAN BINARY SEARCH)**



ABIM MUSTAWA

244107020078

KELAS TI-1B

**PRODI D-IV TEKNIK
INFORMATIKA JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI
MALANG 2025**

1. Percobaan Praktikum

1.1 Praktikum 1 – Searching Menggunakan Sequential Search

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 i = 0; i < listMhs.length; i++) {  
        if (listMhs[i].ipk==cari) {  
            posisi=i;  
            break;  
        }  
    }  
    return posisi;  
}
```

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

```
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, 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.

```

import java.util.Scanner;;
public class MahasiswaDemo01 {
    Run | Debug
    public static void main(String[] args) {
        MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01();
        Scanner sc = new Scanner(System.in);
        int jumlahMhs = 5;

        for (int i = 0; i < jumlahMhs; 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 Mahasiswa01(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:"Manggunakan sequential searching");
        double posisi = list.sequentialSearching(cari);
        int pss = (int) posisi;
        list.tampilPosisi(cari, pss);
        list.tampilDataSearch(cari, pss);
    }
}

```

7. Jalankan dan amati hasilnya.

```

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
-----
Masukkan ipk mahasiswa yang dicari :
IPK : 3.5
Manggunakan 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!

Jawab:

tampilDataSearch menampilkan data mahasiswa yang ditemukan di posisi tersebut, sedangkan tampilPosisi hanya menampilkan informasi index dari elemen yang dicari.

2. Jelaskan fungsi break pada kode program dibawah ini!

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

Jawab:

Digunakan untuk menghentikan perulangan secara ketika kondisi listMhs[i].ipk==cari.

1.2 Praktikum 2 – Searching Menggunakan Binary Search

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

```

// 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.print(s:"IPK : ");
double cari2 = sc.nextDouble();

System.out.println(x:"Manggunakan binary searching");
double posisi2 = list.findBinarySearch(cari2, left:0, jumlahMhs-1);
int pss2 = (int) posisi2;
list.tampilPosisi(cari2, pss2);
list.tampilDataSearch(cari2, pss2);
}

```

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

```

Kelas : 2
IPK : 3.2
-----
Nama : lia
NIM : 333
Kelas : 2
IPK : 3.3
-----
Nama : susi
NIM : 444
Kelas : 2
IPK : 3.5
-----
Nama : anita
NIM : 55
Kelas : 2
IPK : 3.7
-----
-----
Pencarian Data
-----
Masukkan ipk mahasiswa yang dicari :
IPK : 3.7
Manggunakan sequential searching
Data mahasiswa dengan IPK : 3.7 ditemukan pada indeks 4
nim      : 55
nama     : anita
kelas    : 2
ipk      : 3.7
-----
Pencarian Data
-----
Masukkan ipk mahasiswa yang dicari :
IPK : 3.7
Manggunakan binary searching
Data mahasiswa dengan IPK : 3.7 ditemukan pada indeks 4
nim      : 55
nama     : anita
kelas    : 2
ipk      : 3.7

```

Pertanyaan

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

Jawab:

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

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

Jawab:

```
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);  
}
```

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

Jawab:

Program akan tetap berjalan, tetapi hasilnya tidak akurat atau bahkan gagal,

4. Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (missal : 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 seach agar hasilnya sesuai

Jawab:

```
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.

Jawab:

```
public class MahasiswaBerprestasi01 {
    Mahasiswa01[] listMhs;
    int idx = 0;

    public MahasiswaBerprestasi01(int jumlah){
        listMhs = new Mahasiswa01[jumlah];
    }

    Scanner sc = new Scanner(System.in);
    System.out.print(s:"Masukkan jumlah mahasiswa: ");
    int jumlahMhs = sc.nextInt();
    sc.nextLine();
    MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01(jumlahMhs);
```

2. Latihan Praktikum

2.1 Latihan 1

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

- PencarianDataSequential : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.
- PencarianDataBinary : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.
- Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

```
int sequentialSearching(String cari) {
    int posisi = -1;
    int jumlah = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(cari)) {
            if (posisi == -1) posisi = i;
            jumlah++;
        }
    }
    if (posisi != -1) {
        System.out.println("Data dosen dengan nama \" " + cari + "\" ditemukan pada indeks " + posisi);
        for (int i = 0; i < idx; i++) {
            if (dataDosen[i].nama.equalsIgnoreCase(cari)) {
                dataDosen[i].tampil();
            }
        }
        if (jumlah > 1) {
            System.out.println(x:"PERINGATAN: Ditemukan lebih dari satu dosen dengan nama yang sama!");
        }
    } else {
        System.out.println("Data dosen dengan nama \" " + cari + "\" tidak ditemukan.");
    }
    return posisi;
}
```



```

int findBinarySearch(int cari, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;
        if (dataDosen[mid].usia == cari) {
            return mid;
        } else if (dataDosen[mid].usia > cari) {
            return findBinarySearch(cari, left, mid - 1);
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}

void tampilDataSearch(int usia, int pos) {
    if (pos != -1) {
        int jumlah = 0;
        for (int i = 0; i < idx; i++) {
            if (dataDosen[i].usia == usia) {
                dataDosen[i].tampil();
                jumlah++;
            }
        }
        if (jumlah > 1) {
            System.out.println(x:"PERINGATAN: Ditemukan lebih dari satu dosen dengan usia yang sama!");
        }
    } else {
        System.out.println("Data dosen dengan usia " + usia + " tidak ditemukan");
    }
}

```

```

        break;
    case 5:
        System.out.print(s:"Masukkan nama dosen yang dicari: ");
        String cariNama = sc.nextLine();
        daftarDosen.sequentialSearching(cariNama);
        break;
    case 6:
        System.out.print(s:"Masukkan usia dosen yang dicari: ");
        int cariUsia = sc.nextInt();
        sc.nextLine();
        daftarDosen.findBinarySearch(cariUsia, left:0, daftarDosen.idx - 1);
        break;

```