

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 7 SEARCHING**



RANI MIFTAHUS SAÁDAH

244107020057

TI_1E

PROGRAM STUDI D_IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

PRAKTIKUM 2025

7.1.1 SEARCHING/ PENCARIAN MENGGUNAKAN ALGORITMA SEQUENTIAL SEARCH

```
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" );
    }
}

}
```

```

import java.util.Scanner;
public class MahasiswaDemo25 {
    public static void main(String[] args) {
        MahasiswaBerprestasi25 list = new MahasiswaBerprestasi25();
        Scanner sc = new Scanner(System.in);
        int jumMhs=5;

        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: ");
            String ip = sc.nextLine();
            Double ipk = Double.parseDouble(ip);
            System.out.println("_____");
            list.tambah(new Mahasiswa25(nim, nama, kelas, ipk));
        }

        list.tampil();
        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();

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

```

Masukkan data mahasiswa ke-1

NIM: 111

Nama: aldi

Kelas: 2

IPK: 3.0

Masukkan data mahasiswa ke-2

NIM: 222

Nama: ray

Kelas: 2

IPK: 3.9

Masukkan data mahasiswa ke-3

NIM: 333

Nama: mohan

Kelas: 2

IPK: 3.7

Masukkan data mahasiswa ke-4

NIM: 444

Nama: qaila

Kelas: 2

IPK: 3.5

Masukkan data mahasiswa ke-5

NIM: 555

Nama: petok

Kelas: 2

IPK: 3.4

NIM: 111

Nama: aldi

Kelas: 2

IPK: 3.0

NIM: 222

Nama: ray

Kelas: 2

IPK: 3.9

NIM: 333

Nama: mohan

Kelas: 2

IPK: 3.7

NIM: 444

Nama: qaila

Kelas: 2

IPK: 3.5

NIM: 555

Nama: petok

Kelas: 2

IPK: 3.4

```
-----
pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,4
menggunakan sequential searching
data mahasiswa dengan IPK : 3.4 ditemukan pada indeks 4
IPK: 3.4
-----
```

7.1.2 PERTANYAAN

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerpretasi25:

- *tampilDataSearch digunakan untuk menampilkan data lengkap dari mahasiswa yang dicari berdasarkan kriteria tertentu (misalnya berdasarkan IPK). Biasanya akan menampilkan: nama, NIM, IPK, atau data lengkap lainnya dari mahasiswa tersebut.*
- *tampilPosisi digunakan untuk menampilkan posisi (indeks array) dari data mahasiswa yang dicari. Hanya menampilkan posisi di dalam array, bukan data lengkapnya.*

2. Jelaskan fungsi break pada kode program dibawah ini!

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

digunakan untuk menghentikan loop agar program tidak melakukan pencarian lebih lanjut setelah menemukan data yang sesuai.

7.2.1 SEARCHING/ PENACARIAN MENGGUAKAN BINARY SEARCH

```
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;
}
```

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

}
```

```

-----
pencarian data
-----
masukkan ipk mahasiswa yang dicari:
IPK: 3,5
-----
menggunakan sequential searching
-----
data mahasiswa dengan IPK : 3.5 ditemukan pada indeks 2
nim      : 333
nama     : miftah
kelas    : 2
ipk      : 3.5
-----
menggunakan binary search
-----
data mahasiswa dengan IPK : 3.5 ditemukan pada indeks 2
nim      : 333
nama     : miftah
kelas    : 2
ipk      : 3.5
PS D:\Semester 2\PRAKASD\PRAKTIKUM_ASD>

```

7.2.2 PERTANYAAN

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

```

int mid;
if (right >= left){
    mid = (left + right)/2;
}

```

menghitung nilai tengah array untuk memecah data menjadi dua bagian dalam binary search.

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

```

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;

```

menentukan bagian array mana yang akan diperiksa selanjutnya berdasarkan hasil perbandingan IPK.

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

Mengapa demikian?

Program bisa berjalan, tapi hasil pencarian akan salah atau tidak ditemukan, karena Binary Search membutuhkan data dalam keadaan urut.

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?

Tidak sesuai karena algoritma mengasumsikan urutan ascending.

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

> ubah arah untuk urutan descending

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

```
public static void main(String[] args) {
    MahasiswaBerprestasi25 list = new MahasiswaBerprestasi25();
    Scanner sc = new Scanner(System.in);
    System.out.print(s:"Masukkan jumlah mahasiswa: ");
    int jumMhs = sc.nextInt();
    sc.nextLine();
}
```

```
Masukkan jumlah mahasiswa: 5
Masukkan data mahasiswa ke-1
NIM: █
```


LATIHAN

```
int PencarianDataSequential(String nama) {
    int jumlahDitemukan = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(nama)) {
            System.out.println("Dosen ditemukan pada index ke-" +
i);
                dataDosen[i].tampil();
                jumlahDitemukan++;
        }
    }
    if (jumlahDitemukan == 0) {
        System.out.println("Data dosen dengan nama '" + nama + "'
tidak ditemukan.");
    } else if (jumlahDitemukan > 1) {
        System.out.println("Peringatan: Ditemukan lebih dari 1 data
dengan nama tersebut!");
    }
    return jumlahDitemukan;
}
```

```

int PencarianDataBinary(int usiaCari) {
    bubbleSort();
    int low = 0, high = idx - 1;
    int jumlahDitemukan = 0;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (dataDosen[mid].usia == usiaCari) {
            int kiri = mid, kanan = mid;
            while (kiri - 1 >= 0 && dataDosen[kiri - 1].usia ==
usiaCari) kiri--;
            while (kanan + 1 < idx && dataDosen[kanan + 1].usia ==
usiaCari) kanan++;

            for (int i = kiri; i <= kanan; i++) {
                System.out.println("Dosen ditemukan pada index ke-"
+ i);

                dataDosen[i].tampil();
                jumlahDitemukan++;
            }

            if (jumlahDitemukan > 1) {
                System.out.println("Peringatan: Ditemukan lebih dari
1 data dengan usia tersebut!");
            }
            return jumlahDitemukan;
        } else if (usiaCari < dataDosen[mid].usia) {
            high = mid - 1;
        } else {
            low = mid + 1;
        }
    }
    System.out.println("Data dosen dengan usia " + usiaCari + "
tidak ditemukan.");
    return 0;
}
}

```

```

import java.util.Scanner;

public class dosenDemo25 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        dataDosen25 data = new dataDosen25();

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampil Data Dosen");
            System.out.println("3. Sorting ASC (Usia Muda ke Tua)");
            System.out.println("4. Sorting DSC (Usia Tua ke Muda)");
            System.out.println("5. Sorting DSC (Insertion Sort)");
            System.out.println("6. Pencarian Nama Dosen (Sequential Search)");
            System.out.println("7. Pencarian Usia Dosen (Binary Search)");
            System.out.println("8. Keluar");
            System.out.print("Pilih: ");
            int pilih = input.nextInt();
            input.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Masukkan kode: ");
                    String kode = input.nextLine();
                    System.out.print("Masukkan nama: ");
                    String nama = input.nextLine();
                    System.out.print("Masukkan jenis kelamin (true untuk Laki-laki, false
untuk Perempuan): ");
                    boolean jk = input.nextBoolean();
                    System.out.print("Masukkan usia: ");
                    int usia = input.nextInt();
                    data.tambah(new Dosen25(kode, nama, jk, usia));
                    break;
                case 2:
                    data.tampil();
                    break;
                case 3:
                    data.bubbleSort();
                    System.out.println("-----");
                    System.out.println("Data dosen setelah sorting (ASC).\n");
                    data.tampil();
                    break;
                case 4:
                    data.SelectionSort();
                    System.out.println("-----");
                    System.out.println("Data dosen setelah sorting (DSC - Selection
Sort).\n");
                    data.tampil();
                    break;
                case 5:
                    data.insertionSort();
                    System.out.println("-----");
                    System.out.println("Data dosen setelah sorting (DSC - Insertion
Sort).\n");
                    data.tampil();
                    break;
                case 6:
                    System.out.print("Masukkan nama yang dicari: ");
                    String cariNama = input.nextLine();
                    data.PencarianDataSequential(cariNama);
                    break;
                case 7:
                    System.out.print("Masukkan usia yang dicari: ");
                    int cariUsia = input.nextInt();
                    data.PencarianDataBinary(cariUsia);
                    break;
                case 8:
                    System.out.println("Keluar dari program.");
                    input.close();
                    return;
                default:
                    System.out.println("Pilihan tidak valid!");
            }
        }
    }
}

```

Menu:

1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Sorting DSC (Insertion Sort)
6. Pencarian Nama Dosen (Sequential Search)
7. Pencarian Usia Dosen (Binary Search)
8. Keluar

Pilih: 2

kode: 001, nama: wahyu, jenis kelamin: laki laki, usia: 30
kode: 002, nama: rudi, jenis kelamin: laki laki, usia: 35
kode: 003, nama: fida, jenis kelamin: perempuan, usia: 31

Pilih: 3

Data dosen setelah sorting (ASC).

kode: 001, nama: wahyu, jenis kelamin: laki laki, usia: 30
kode: 003, nama: fida, jenis kelamin: perempuan, usia: 31
kode: 002, nama: rudi, jenis kelamin: laki laki, usia: 35

Pilih: 4

Data dosen setelah sorting (DSC - Selection Sort).

kode: 002, nama: rudi, jenis kelamin: laki laki, usia: 35
kode: 003, nama: fida, jenis kelamin: perempuan, usia: 31
kode: 001, nama: wahyu, jenis kelamin: laki laki, usia: 30

Pilih: 5

Data dosen setelah sorting (DSC - Insertion Sort).

kode: 002, nama: rudi, jenis kelamin: laki laki, usia: 35
kode: 003, nama: fida, jenis kelamin: perempuan, usia: 31
kode: 001, nama: wahyu, jenis kelamin: laki laki, usia: 30

Pilih: 6

Masukkan nama yang dicari: rudi

Dosen ditemukan pada index ke-0

kode: 002, nama: rudi, jenis kelamin: laki laki, usia: 35

Pilih: 7

Masukkan usia yang dicari: 31

Dosen ditemukan pada index ke-1

kode: 003, nama: fida, jenis kelamin: perempuan, usia: 31

