

LAPORAN HASIL PRAKTIKUM

Algoritma Dan Struktur Data

Searching



NAMA : Handino Asa Galih r

NIM : 244107020237

KELAS : 1-E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLINEMA

2025

Searching / Pencarian Menggunakan Algoritma Sequential Search

Kode Program Class MahasiswaBerprestasi12

```
package jobsheet7;

public class MahasiswaBerprestasi12 {

    Mahasiswa12 [] listMhs;

    int idx = 0;

    public MahasiswaBerprestasi12 (int jumMhs) {

        listMhs = new Mahasiswa12[jumMhs];

    }

    void tambah (Mahasiswa12 m) {

        if (idx < listMhs.length) {

            listMhs [idx] = m;

            idx++;

        } else {

            System.out.println ("Data sudah penuh");

        }

    }

    void tampil() {

        for (Mahasiswa12 m:listMhs) {

            if (m != null) {

                m.tampilInformasi();

                System.out.println("-----

");

            }

        }

    }

}
```

```

void bubbleSort() {
    for (int i = 0; i < listMhs.length-1; i++) {
        for (int j = 1; j < listMhs.length-i; j++) {
            if (listMhs[j].ipk > listMhs[j-1].ipk) {
                Mahasiswa12 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}

void selectionSort() {
    for (int i = 0; i < listMhs.length-1; i++) {
        int idxMin = i;
        for (int j = i+1; j < listMhs.length; j++) {
            if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                idxMin = j;
            }
        }
        Mahasiswa12 tmp = listMhs[idxMin];
        listMhs [idxMin] = listMhs[i];
        listMhs [i] = tmp;
    }
}

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa12 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk < temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}

```

```

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");
    }
}

```

Kode Program Class MahasiswaDemo12

```
package jobsheet7;

import java.util.Scanner;

public class MahasiswaDemo12 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan jumlah mahasiswa : ");

        int jumMhs = sc.nextInt();

        MahasiswaBerprestasi12 list = new
        MahasiswaBerprestasi12(jumMhs);

        for (int i = 0; i < jumMhs; i++) {

            System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));

            System.out.print("NIM      : ");

            String nim = sc.next();

            System.out.print("Nama      : ");

            String nama = sc.next();

            System.out.print("Kelas    : ");

            String kelas = sc.next();

            System.out.print("IPK      : ");

            double ipk = sc.nextDouble();

            System.out.println("-----");

            list.tambah(new Mahasiswa12(nim, nama, kelas, ipk));

        }

        list.tampil();

        System.out.println("-----");

        System.out.println("Pencarian Data");

        System.out.println("-----");

        System.out.print("Masukkan ipk mahasiswa yang dicari: ");

        System.out.print("IPK: ");

        double cari = sc.nextDouble();

        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);


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);

}

}

```

Output

Masukkan Data Mahasiswa ke-1	Nama : adi
NIM : 111	NIM : 111
Nama : adi	Kelas : 2
Kelas : 2	IPK : 3.6
IPK : 3,6	-----
Masukkan Data Mahasiswa ke-2	Nama : tio
NIM : 222	NIM : 222
Nama : tio	Kelas : 2
Kelas : 2	IPK : 3.8
IPK : 3,8	-----
Masukkan Data Mahasiswa ke-3	Nama : ila
NIM : 333	NIM : 333
Nama : ila	Kelas : 2
Kelas : 2	IPK : 3.0
IPK : 3	-----
Masukkan Data Mahasiswa ke-4	Nama : lia
NIM : 444	NIM : 444
Nama : lia	Kelas : 2
Kelas : 2	IPK : 3.5
IPK : 3,5	-----
Masukkan Data Mahasiswa ke-5	Nama : fia
NIM : 555	NIM : 555
Nama : fia	Kelas : 2
Kelas : 2	IPK : 3.3
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

```

Pertanyaan

1. Jelaskan perbedaan method tampilDataSearch dan tampilPosisi pada Class MahasiswaBerprestasi!

Metode tampilPosisi menampilkan posisi atau indeks data mahasiswa dengan IPK tertentu dalam array, sementara tampilDataSearch menampilkan informasi lengkap tentang mahasiswa yang ditemukan, seperti NIM, Nama, Kelas, dan IPK. Perbedaan utamanya adalah tampilPosisi hanya menunjukkan posisi, sedangkan tampilDataSearch memberikan detail lengkap tentang mahasiswa.

2. Jelaskan fungsi break pada kode program di bawah ini!

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

Fungsi break pada kode program tersebut digunakan untuk menghentikan eksekusi loop secara langsung Ketika kondisi tertentu terpenuhi. Dalam konteks kode yang diberikan, break berfungsi untuk menghentikan pencarian setelah ditemukan mahasiswa yang memiliki IPK yang dicari. Ketika kondisi `listMhs[j].ipk == cari` bernilai true (artinya IPK pada indeks j sama dengan IPK yang dicari), maka variable posisi akan diisi dengan nilai j, yang menandakan bahwa data ditemukan pada indeks tersebut. Kemudian, perintah break akan menghentikan iterasi loop lebih lanjut karena pencarian sudah selesai.

Searching / Pencarian Menggunakan Binary Search

Penambahan Kode Program Class MahasiswaBerprestasi12

```
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, mid+1, right);  
        }  
    }
```

```

        else {
            return findBinarySearch(cari, left, mid-1);
        }
    }
    return -1;
}
}

```

Penambahan Kode Program Class MahasiswaDemo12

```

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);

```

Pertanyaan

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

```

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, mid+1, right);
    }
    else {
        return findBinarySearch(cari, left, mid-1);
    }
}

```


Pada kode program yang menggunakan metode binary search di dalam class MahasiswaBerprestasi05, proses divide terjadi di bagian perhitungan nilai mid dan pemanggilan rekursif ke bagian kiri atau kanan array yang terjadi di baris $\text{mid} = (\text{left} + \text{right}) / 2$, di mana array dibagi menjadi dua bagian dengan mencari nilai tengah (mid) dari rentang indeks left dan right.

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

```
if (cari == listMhs[mid].ipk) {  
    return (mid);  
}
```

Proses conquer pada binary search terjadi ketika nilai tengah (`listMhs[mid].ipk`) sama dengan nilai yang dicari (`cari`). Jika kondisi ini terpenuhi, maka pencarian selesai dan indeks mid yang menunjukkan posisi elemen yang ditemukan dikembalikan. Hal ini menandakan bahwa pencarian telah berhasil, dan elemen yang dicari telah ditemukan, yang merupakan langkah conquer setelah array dibagi menjadi dua bagian pada proses divide.

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

Program masih bisa berjalan, tetapi binary search tidak akan berfungsi dengan benar jika data IPK tidakurut. Binary search memerlukan data yang terurut untuk membagi pencarian dengan efektif. Tanpa pengurutan, pencarian akan menghasilkan hasil yang tidak akurat. Jadi, data perlu diurutkan terlebih dahulu agar binary search bisa bekerja dengan benar.

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. Jika IPK yang dimasukkan dalam urutan dari terbesar ke terkecil (misalnya 3.8, 3.7, 3.5, 3.4, 3.2) dan elemen yang dicari adalah 3.2, maka hasil dari binary search yang ada pada program saat ini tidak akan sesuai. Ini karena binary search hanya berfungsi dengan baik pada data yang terurut dalam urutan kecil ke besar (ascending order), sedangkan dalam kasus ini data disusun besar ke kecil (descending order). memperbaiki hal ini, kita perlu memodifikasi fungsi `findBinarySearch` seperti kode dibawah ini :

```

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

```

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

Penambahan Kode Program Class MahasiswaDemo12

```

System.out.print(s:"Masukkan jumlah mahasiswa : ");
int jumMhs = sc.nextInt();
MahasiswaBerprestasi12 list = new MahasiswaBerprestasi12(jumMhs);

```

Penambahan Kode Program Class MahasiswaBerprestasi12

```

public class MahasiswaBerprestasi12 {
    Mahasiswa12 [] listMhs;
    int idx = 0;

    public MahasiswaBerprestasi12 (int jumMhs) {
        listMhs = new Mahasiswa12[jumMhs];
    }

```

```

void tampil() {
    for (Mahasiswa12 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
-----
```

Latihan Praktikum

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!

Penambahan Kode Program Class DosenDemo12

```
System.out.print("Masukkan jumlah dosen : ");
int jumlah = input.nextInt();

DataDosen12 dosen = new DataDosen12(jumlah);
```

```

System.out.println("-----");
System.out.println("Data dosen sebelum sorting ");
System.out.println("-----");
dosen.tampil();

System.out.println("-----");
System.out.println("Searching menggunakan Sequential Search");
System.out.println("-----");
System.out.print("Masukkan nama dosen yang dicari : ");
String cariNama = input.nextLine();
dosen.pencarianDataSequential(cariNama);

System.out.println("-----");
System.out.println("Searching menggunakan Binary Search");
System.out.println("-----");
System.out.print("Masukkan usia yang dicari : ");
int cari = input.nextInt();
dosen.SortingASC();
dosen.pencarianDataBinary(cari, 0, jumlah-1);

```

Penambahan Kode Program Class DataDosen12

```

Dosen12[] dataDosen;
int idx;

public DataDosen12(int var1) {
    this.dataDosen = new Dosen12[var1];
}

```

```

void pencarianDataSequential(String var1) {
    int var2 = 0;

    for(int var3 = 0; var3 < this.dataDosen.length; ++var3) {
        if (this.dataDosen[var3].nama.equalsIgnoreCase(var1)) {
            String var10001 = this.dataDosen[var3].nama;
            System.out.println("Data dosen dengan nama : " + var10001 + " ditemukan pada");
            System.out.println("Dengan detail data dosen : ");
            Dosen12 var4 = this.dataDosen[var3];
            System.out.println("Kode          : " + var4.kode);
            var4 = this.dataDosen[var3];
            System.out.println("Nama          : " + var4.nama);
            if (this.dataDosen[var3].jenisKelamin) {
                System.out.println("Jenis Kelamin : Wanita");
            } else {
                System.out.println("Jenis Kelamin : Pria");
            }

            var4 = this.dataDosen[var3];
            System.out.println("Usia          : " + var4.usia);
            System.out.println();
            ++var2;
        }
    }

    if (var2 == 0) {
        System.out.println("Data tidak ditemukan.");
    } else {
        System.out.println("=====");
        System.out.println("PERINGATAN : Data ditemukan sebanyak " + var2);
        System.out.println("=====");
        if (var2 > 1) {
            System.out.println("PERINGATAN: Terdapat lebih dari satu data dosen dengan na");
        }
    }
}

```

```

void pencarianDataBinary(int var1, int var2, int var3) {
    int var5 = 0;
    int[] var6 = new int[this.dataDosen.length];

    while(var3 >= var2) {
        int var4 = (var2 + var3) / 2;
        if (var1 == this.dataDosen[var4].usia) {
            var6[var5] = var4;
            ++var5;

            int var7;
            for(var7 = var4 - 1; var7 >= var2 && this.dataDosen[var7].usia == var1;
                var6[var5] = var7;
                ++var5;
            }

            for(var7 = var4 + 1; var7 <= var3 && this.dataDosen[var7].usia == var1;
                var6[var5] = var7;
                ++var5;
            }

            System.out.println("=====");
            System.out.println("PERINGATAN : Data ditemukan sebanyak " + var5);
            System.out.println("=====");

            for(int var8 = 0; var8 < var5; ++var8) {
                int var9 = var6[var8];
                int var10001 = this.dataDosen[var9].usia;
                System.out.println("Data dosen dengan usia : " + var10001 + " ditemu");
                System.out.println("Dengan detail data dosen : ");
                Dosen12 var10 = this.dataDosen[var9];
                System.out.println("Kode          : " + var10.kode);
                var10 = this.dataDosen[var9];
                System.out.println("Nama          : " + var10.nama);
                if (this.dataDosen[var9].jenisKelamin) {
                    System.out.println("Jenis Kelamin : Wanita");
                } else {
                    System.out.println("Jenis Kelamin : Pria");
                }
            }
        }
    }
}

```

Output

```
Masukkan jumlah dosen : 3
Masukkan jumlah data dosen ke-1
Masukkan Kode       : 111
Masukkan Nama       : Athaulla
Masukkan Jenis Kelamin : Pria
Masukkan Usia       : 23
```

```
Masukkan jumlah data dosen ke-2
Masukkan Kode       : 222
Masukkan Nama       : Hafizh
Masukkan Jenis Kelamin : Pria
Masukkan Usia       : 23
```

```
Masukkan jumlah data dosen ke-3
Masukkan Kode       : 333
Masukkan Nama       : Zahra
Masukkan Jenis Kelamin : Wanita
Masukkan Usia       : 30
```

Data dosen sebelum sorting

```
Kode       : 111
Nama       : Athaulla
Jenis Kelamin : Pria
Usia       : 23
-----
```

```
Kode       : 222
Nama       : Hafizh
Jenis Kelamin : Pria
Usia       : 23
-----
```

```
Kode       : 333
Nama       : Zahra
Jenis Kelamin : Wanita
Usia       : 30
-----
```

Searching menggunakan Sequential Search

```
Masukkan nama dosen yang dicari : Hafizh
Data dosen dengan nama : Hafizh ditemukan pada indeks 1
Dengan detail data dosen :
Kode       : 222
Nama       : Hafizh
Jenis Kelamin : Pria
Usia       : 23
```

=====
PERINGATAN : Data ditemukan sebanyak 1
=====

Searching menggunakan Binary Search

```
Masukkan usia yang dicari : 23
```

=====
PERINGATAN : Data ditemukan sebanyak 2
=====

```
Data dosen dengan usia : 23 ditemukan pada indeks 1
Dengan detail data dosen :
Kode       : 222
Nama       : Hafizh
Jenis Kelamin : Pria
Usia       : 23
```

```
Data dosen dengan usia : 23 ditemukan pada indeks 0
Dengan detail data dosen :
Kode       : 111
Nama       : Athaulla
Jenis Kelamin : Pria
Usia       : 23
```

PERINGATAN: Terdapat lebih dari satu data dosen dengan usia tersebut.