

Laporan Hasil Praktikum
Algoritma dan Struktur Data
Jobsheet 7 Searching



Muhammad Firman Aditiasmara

244107020094

TI-1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Menjelaskan mengenai algoritma Searching.
2. Membuat dan mendeklarasikan struktur algoritma Searching.
3. Menerapkan dan mengimplementasikan algoritma Searching

2. Praktikum

2.1 Percobaan 1: Searching / Pencarian Menggunakan Algoritma Sequential Search

2.1.1 Langkah-langkah Percobaan

1. Menambahkan method sequentialSearch pada clas MahasiswaBerprestasi18

```
int sequentialSearch(double cari) {  
    int posisi = -1;  
    for (int j = 0; j < listMhs.length; j++) {  
        if (listMhs[j].ipk == cari) {  
            posisi = j;  
            break;  
        }  
    }  
    return posisi;  
}
```

2. Membuat method tampilPosisi bertipe void pada class MahasiswaBerprestasi18
3. Membuat method tampilDataSearch berti[e void pada class

```
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");  
    }  
}
```

MahasiswaBerprestasi18

```

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");
    }
}
}

```

4. Menambahkan kode program pada class MahasiswaDemo18

```

Scanner input = 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 = input.nextLine();
    System.out.print("Nama      : ");
    String nama = input.nextLine();
    System.out.print("Kelas    : ");
    String kelas = input.nextLine();
    System.out.print("IPK      : ");
    String ip = input.nextLine();
    Double ipk = Double.parseDouble(ip);
    System.out.println("-----");

    list.tambah(new Mahasiswa18(nim, nama, kelas, ipk));
}

list.tampil();

// melakukan pencarian data Sequential
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 = input.nextDouble();

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

```

2.1.2 Verifikasi Hasil Percobaan

```
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
-----

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
```

```

-----
Nama      : fia
NIM       : 555
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

```

2.1.3 Pertanyaan

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

2. Jelaskan fungsi **break** pada kode program dibawah ini!

```

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

```

Jawab : pada kode tersebut `int min = i;` menyimpan nilai terkecil yang ditemukan, Ketika perulangan dilanjutkan, jika menemukan nilai minimum lagi maka `min =j;` akan diperbarui

2.2 Percobaan 2: Searching / Pencarian Menggunakan Binary Search

2.2.1 Langkah-langkah Percobaan

1. Menambahkan method findBinarySearch bertipe integer pada class MahasiswaBerprestasi18.

```
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. Memanggil method findBinarySearch di kelas MahasiswaDemo18, kemudian memanggil method tampilPosisi dan tampilDataSearch

```
// melakukan pencarian data Binary
System.out.println("-----
-----");
System.out.println("Pencarian data");
System.out.println("-----
-----");

System.out.println("Masukkan ipk mahasiswa yang
dicari: ");
System.out.print("IPK: ");
cari = input.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);
```

2.2.2 Verifikasi Hasil Percobaan

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

Nama : adi
NIM : 111
Kelas : 2
IPK : 3.1

Nama : ila
NIM : 222
Kelas : 2
IPK : 3.2

Nama : lia
NIM : 333
Kelas : 2
IPK : 3.3

Nama : susi
NIM : 444
Kelas : 2
IPK : 3.5

```

IPK      : 3.5
-----
Nama     : susi
NIM      : 444
Kelas   : 2
IPK      : 3.5
-----
Nama     : anita
NIM      : 555
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
adid@adid-IdeaPad-Gaming-3-15ACH6:/media/adid/Data dan Hiburan/Data Tugas/SA

```

2.2.3 Pertanyaan

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

Jawab : Proses divide terdapat pada kode program berikut, proses divide sendiri terjadi dengan memisahkan dua bagian dalam permasalahan program sehingga akan menemukan nilai tengah (mid)

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

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

Jawab :

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

Kode program tersebut merupakan proses conquer sebelah kiri, setelah proses dipecah menjadi dua bagian saat proses divide, maka proses conquer akan menyelesaikan masalah bagian kiri dan kanan

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

Kode diatas merupakan proses conquer bagian kanan

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

Jawab : Jika data IPK tidak dimasukkan secara urut, maka program tidak dapat berjalan dengan benar, maka nilai yang dicari bisa tidak ditemukan padahal nilai tersebut terdapat pada data

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

Jawab : dari ilustrasi tersebut diketahui bahwa data diurutkan dari nilai terbesar ke kecil, yang artinya data tersebut diurutkan secara descending. Maka perlu dilakukan perubahan pada program binary search

```
} else if (listMhs[mid].ipk < cari) {
```

Program binary search hanya perlu merubah tanda <, ini akan membuat pencarian akan mengarah pada nilai yang diinginkan yaitu 3.2

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

Jawab : Modifikasi program untuk menentukan jumlah mahasiswa yang akan diinputkan sesuai dengan inputan pengguna

```
System.out.println("Masukkan jumlah mahasiswa: ");  
  
int jumMhs = input.nextInt();
```

Dengan perubahan diatas, maka jumlah data banyaknya mahasiswa akan dapat ditentukan oleh inputan dari pengguna

2.3 Latihan Praktikum

1 Pada Latihan praktikum pertemuan sebelumnya pada Jobsheet 6 yang terdapat 3 class yaitu **Dosen**<no presensi>, **DataDosen**<no presensi> , dan **DosenDemo**<no presensi>, tambahkan method:

- PencarianDataSequential<no presensi> : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.
- PencarianDataBinary<no presensi> : 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

Kode Program

1. Kode program Dosen18

```
public class Dosen18 {

    String kode;
    String nama;
    boolean jenisKelamin;
    int umur;

    Dosen18() {

    }

    Dosen18(String kd, String nm, boolean jk, int age) {
        kode = kd;
        nama = nm;
        jenisKelamin = jk;
        umur = age;
    }

    void tampil(){
        System.out.println("Kode\t\t : " + kode);
        System.out.println("Nama\t\t : " + nama);
        System.out.println("Jenis Kelamin\t : " + (jenisKelamin?
"Laki-laki" : "Perempuan"));
        System.out.println("Umur\t\t : " + umur);
    }

}
```

2. Kode Program DosenData18

```
import java.util.Scanner;

public class DosenData18 {

    Scanner scan = new Scanner(System.in);
    Dosen18[] listDosen = new Dosen18[3];
    int idx = 0;

    void tambah(Dosen18 dsn) {
        if (idx >= listDosen.length) {
            System.out.println("Data telah penuh, tidak dapat
menambahkan lebih banyak dosen.");
            return;
        }
        listDosen[idx++] = dsn;
        System.out.println("Dosen berhasil ditambahkan!");
    }

    void input() {
        for (int i = 0; i < listDosen.length; i++) {
            System.out.println("Masukkan data Dosen ke-" + (i + 1));
            Dosen18 dsn = new Dosen18();
            System.out.print("Kode\t\t\t\t\t: ");
            dsn.kode = scan.next();
            System.out.print("Nama\t\t\t\t\t: ");
            dsn.nama = scan.next();
            scan.nextLine(); // Konsumsi newline
            System.out.print("Jenis Kelamin(L/P)\t\t\t: ");
            String jk = scan.next();
            dsn.jenisKelamin = jk.equalsIgnoreCase("L");
            System.out.print("Umur\t\t\t\t\t: ");
            dsn.umur = scan.nextInt();
            tambah(dsn);
        }
    }

    void tampil() {
        for (int i = 0; i < idx; i++) {
            listDosen[i].tampil();
            System.out.println("-----");
        }
    }

    void SortingASC() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - i - 1; j++) {
                if (listDosen[j].umur > listDosen[j + 1].umur) {
                    Dosen18 temp = listDosen[j];
                    listDosen[j] = listDosen[j + 1];
                    listDosen[j + 1] = temp;
                }
            }
        }
    }
}
```

```

void SortingDSC() {
    for (int i = 0; i < idx - 1; i++) {
        for (int j = 0; j < idx - i - 1; j++) {
            if (listDosen[j].umur < listDosen[j + 1].umur) {
                Dosen18 temp = listDosen[j];
                listDosen[j] = listDosen[j + 1];
                listDosen[j + 1] = temp;
            }
        }
    }
}

int PencarianDataSequential18(String cari) {
    for (int i = 0; i < idx; i++) {
        if (listDosen[i].nama.equalsIgnoreCase(cari)) {
            return i;
        }
    }
    return -1;
}

int PencarianDataBinary18(int cariUmur, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;
        if (listDosen[mid].umur == cariUmur) {
            return mid;
        } else if (listDosen[mid].umur > cariUmur) {
            return PencarianDataBinary18(cariUmur, left, mid - 1);
        } else {
            return PencarianDataBinary18(cariUmur, mid + 1, right);
        }
    }
    return -1;
}

void tampilPosisi(String nama, int pos) {
    if (pos != -1) {
        System.out.println("Data dosen dengan nama \"" + nama + "\"
ditemukan pada data ke-" + (pos + 1));
    } else {
        System.out.println("Data dosen dengan nama \"" + nama + "\"
tidak ditemukan.");
    }
}

void tampilDataSearch(String cariNama, int pos) {
    if (pos != -1) {
        System.out.println("Data dosen ditemukan:");
        listDosen[pos].tampil();
    } else {
        System.out.println("Data dosen dengan nama \"" + cariNama +
"\n tidak ditemukan.");
    }
}

```

```

void tampilPosisiBinary(int umur, int pos) {
    int jumlah = 0;

    if (pos != -1) {

        for (int i = 0; i < idx; i++) {
            if (listDosen[i].umur == umur) {
                listDosen[i].tampil();
                System.out.println("-----");
                jumlah++;
            }
        }

        if (jumlah > 1) {
            System.out.println("Ditemukan lebih dari satu dosen
dengan umur tersebut sebanyak " + jumlah + " data.");
        } else {
            System.out.println("Ditemukan 1 data dosen dengan umur
tersebut.");
        }

        } else {
            System.out.println("Data dosen dengan umur tersebut tidak
ditemukan.");
        }
    }

    void tampilDataSearchBinary(int cariUmur, int pos) {
        if (pos != -1) {

            } else {
                System.out.println("Data dosen dengan Umur \"" + cariUmur +
"\n" tidak ditemukan.");
            }
        }
    }
}

```

3. Kode Program DosenMain18

```

import java.util.Scanner;

public class DosenMain18 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        DosenData18 list = new DosenData18();

        // Input data dosen
        list.input();

        // System.out.println();
        // System.out.println("Tampilan Data");
        // System.out.println("-----");

        // System.out.println("Data Asli");
        // list.tampil();

        // System.out.println("Data diurutkan secara ASC (Ascending)");
        // list.SortingASC();
        // list.tampil();

        // System.out.println("Data diurutkan secara DSC
(Descending)");
        // list.SortingDSC();
        // list.tampil();

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

        System.out.print("Masukkan nama dosen yang dicari: ");
        String cari = input.next();

        System.out.print("Masukkan Umur dosen yang dicari: ");
        int cariUmur = input.nextInt();

        System.out.println("\nMenggunakan sequential searching");
        int posisiSequential = list.PencarianDataSequential18(cari);
        list.tampilPosisi(cari, posisiSequential);
        list.tampilDataSearch(cari, posisiSequential);

        System.out.println("\nMenggunakan binary searching");
        list.SortingASC();
        int posisiBin = list.PencarianDataBinary18(cariUmur, 0,
list.idx - 1);
        list.tampilPosisiBinary(cariUmur, posisiBin);
        list.tampilDataSearchBinary(cariUmur, posisiBin);
    }
}

```

Hasil Output Kode Program

Output dengan Panjang array indeks array yaitu 3

```
7/Jul_2023/08:22:50/BIN Dosenmain18
Masukkan data Dosen ke-1
Kode           : DSN001
Nama           : Adid
Jenis Kelamin(L/P) : L
Umur           : 33
Dosen berhasil ditambahkan!
Masukkan data Dosen ke-2
Kode           : DSN002
Nama           : Mara
Jenis Kelamin(L/P) : L
Umur           : 33
Dosen berhasil ditambahkan!
Masukkan data Dosen ke-3
Kode           : DSN003
Nama           : Fara
Jenis Kelamin(L/P) : P
Umur           : 39
Dosen berhasil ditambahkan!
-----
Pencarian data
-----
Masukkan nama dosen yang dicari: Fara
Masukkan Umur dosen yang dicari: 33

Menggunakan sequential searching
Data dosen dengan nama "Fara" ditemukan pada data ke-3
Data dosen ditemukan:
Kode           : DSN003
Nama           : Fara
Jenis Kelamin  : Perempuan
Umur           : 39

Menggunakan binary searching
Kode           : DSN001
Nama           : Adid
Jenis Kelamin  : Laki-laki
Umur           : 33
-----
Kode           : DSN002
Nama           : Mara
Jenis Kelamin  : Laki-laki
Umur           : 33
-----
Ditemukan lebih dari satu dosen dengan umur tersebut sebanyak 2 data.
```