

JOBSHEET VII

SEARCHING

Ka Abi Muhammad R.F./12/TI-1B

7.1. 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.

7.2.1. Langkah-langkah Percobaan Sequential Search

1. Pada pertemuan Jobsheet 7 ini akan menggunakan class Mahasiswa<no Presensi>, MahasiswaBerprestasi<no Presensi>, dan MahasiswaDemo<no presensi> 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<no presensi>. Kemudian Deklarasikan isi method sequentialSearching dengan algoritma pencarian data menggunakan teknik sequential searching.

```
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;  
}
```

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.

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    int jumMhs = 5;

    MahasiswaBerprestasi12 list = new MahasiswaBerprestasi12(jumMhs);

    for (int i = 0; i < jumMhs; i++) {
        System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");
        System.out.print(s:"NIM: ");
        String nim = input.nextLine();
        System.out.print(s:"Nama: ");
        String nama = input.nextLine();
        System.out.print(s:"Kelas: ");
        String kelas = input.nextLine();
        System.out.print(s:"IPK: ");
        double ipk = input.nextDouble();
        input.nextLine();

        Mahasiswa12 m = new Mahasiswa12(nim, nama, kelas, ipk);
        list.tambah(m);
    }
```

```
    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.println(x:"IPK: ");
    double cari = input.nextDouble();

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

7. Jalankan dan amati hasilnya.

7.3.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini.

```
dhat.java\jdt_ws\Jobsheet 7_88ccf3f6\bin' 'MahasiswaDemo12'
Masukkan data mahasiswa ke-1:
NIM: 111
Nama: ado
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: ado
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
menggunakan sequential searching
data mahasiswa dengan IPK :3.5 ditemukan pada indeks 3
nim      : 444
nama     : lia
kelas    : 2
ipk      : 3.5
PS C:\Users\kaabi\OneDrive\Desktop\Praktikum ASD\Jobsheet 7> █
```

7.2.3. Pertanyaan

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class

MahasiswaBerprestasi!

tampilPosisi

Tujuan:

Menampilkan indeks posisi dari data mahasiswa yang dicari berdasarkan IPK.

Penjelasan:

- Fungsi ini hanya memberitahukan apakah data ditemukan dan di indeks berapa.
- Tidak menampilkan detail mahasiswa seperti nama, NIM, kelas, dsb.

tampilDataSearch

Tujuan:

Menampilkan **detail informasi mahasiswa** pada posisi (pos) tertentu jika ditemukan berdasarkan **IPK**.

Penjelasan:

- Fungsi ini memberikan informasi lengkap mahasiswa yang memiliki IPK sesuai nilai x.
- Digunakan setelah kita tahu **posisi pasti dari mahasiswa tersebut** (misal hasil dari pencarian).

2. Jelaskan fungsi break pada kode program dibawah ini!

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

}

1. Perulangan for akan mengecek satu per satu mahasiswa di listMhs.
2. Jika ditemukan mahasiswa dengan IPK yang **sama dengan nilai yang dicari (cari)**, maka:
 - o Posisi (posisi) disimpan.
 - o break; akan **langsung menghentikan perulangan** meskipun belum sampai akhir array.

7.3. Searching / Pencarian Menggunakan Binary Search

7.3.1. Langkah-langkah Percobaan 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

```

    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.println(x:"IPK: ");
    double cari = input.nextDouble();
    System.out.println(x:"-----");
    System.out.println(x:"menggunakan binary search");
    System.out.println(x:"-----");
    double posisi2 = list.findBinarySearch(cari, left:0, jumMhs-1);
    int pss2 = (int) posisi2;
    list.tampilPosisi(cari, pss2);
    list.tampilDataSearch(cari, pss2);
}

```

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

7.3.2. 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: anota
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
-----
Nama: anota
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     : anota
kelas    : 2
ipk      : 3.7
PS C:\Users\kaabi\OneDrive\Desktop\Praktikum ASD\Jobsheet 7>

```

7.3.3.

Pertanyaan

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

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

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);  
}
```

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

Bisa berjalan, tapi tergantung fitur yang diterapkan, hal ini tidak akan bekerja benar kalau misal kita pakai binary search, Karena binary search mengandalkan pemotongan data di tengah untuk memutuskan arah pencarian (kiri atau kanan), dan itu hanya valid jika data sudah terurut (baik ascending atau descending).

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

```
3.2  
-----  
menggunakan binary search  
-----  
data 3.2tidak ditemukan  
Data mahasiswa dengan IPK 3.2tidak ditemukan  
PS C:\Users\kaabi\OneDrive\Desktop\Praktikum ASD\Jobsheet 7  
>
```

Program tetap pakai logika kalau data < mid, maka cari ke kiri, padahal arahnya sudah terbalik. Maka pencariannya jadi salah arah.

Solusinya mengubah kondisi pada if (awalnya >)

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



```
IPK: 3.1
-----
-----
Pencarian data
-----
Masukkan ipk mahasiswa yang dicari:
IPK:
3.2
-----
menggunakan binary search
-----
data mahasiswa dengan IPK :3.2 ditemukan pada indeks 3
nim      : r
nama     : r
kelas    : 2
ipk      : 3.2
PS C:\Users\kaabi\OneDrive\Desktop\Praktikum ASD\Jobsheet 7
> |
```

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

TUGAS

```
public class Dosen12 {  
    String kode;  
    String nama;  
    boolean jenisKelamin;  
    int usia;  
  
    public Dosen12(String kd, String name, boolean jk, int age) {  
        this.kode = kd;  
        this.nama = name;  
        this.jenisKelamin = jk;  
        this.usia = age;  
    }  
  
    public void tampil() {  
        System.out.println("Kode      : " + kode);  
        System.out.println("Nama      : " + nama);  
        System.out.println("Jenis Kelamin: " + (jenisKelamin ? "Laki-laki" : "Perempuan"));  
        System.out.println("Usia      : " + usia);  
        System.out.println(x:"-----");  
    }  
}
```

```

import java.util.Scanner;
public class DosenMain12 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen12 listDosen = new DataDosen12();

        while (true) {
            System.out.println(x: "\n===== MENU DATA DOSEN =====");
            System.out.println(x: "1. Tambah Data Dosen");
            System.out.println(x: "2. Tampilkan Data Dosen");
            System.out.println(x: "3. Sorting ASC (Bubble Sort)");
            System.out.println(x: "4. Sorting DSC (Bubble Sort)");
            System.out.println(x: "5. Sorting ASC (Insertion Sort)");
            System.out.println(x: "6. Sequential Search (by Nama)");
            System.out.println(x: "7. Binary Search (by Usia)");
            System.out.println(x: "8. Keluar");
            System.out.print(s: "Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.println(x: "\nMasukkan Data Dosen:");
                    System.out.print(s: "Kode      : ");
                    String kode = sc.nextLine();
                    System.out.print(s: "Nama      : ");
                    String nama = sc.nextLine();
                    System.out.print(s: "Jenis Kelamin (L/P): ");
                    char jk = sc.next().charAt(index:0);
                    boolean jenisKelamin = (jk == 'L' || jk == 'P');
                    System.out.print(s: "Usia      : ");
                    int usia = sc.nextInt();
                    sc.nextLine();

                    Dosen12 dsn = new Dosen12(kode, nama, jenisKelamin, usia);
                    listDosen.tambah(dsn);
                    break;
            }
        }
    }
}

```

```

case 2:
System.out.println(x:"\n===== DATA DOSEN =====");
listDosen.tampil();
break;

case 3:
listDosen.SortingASC();
System.out.println(x:"\nData setelah sorting ASCENDING:");
listDosen.tampil();
break;

case 4:
listDosen.SortingDSC();
System.out.println(x:"\nData setelah sorting DESCENDING:");
listDosen.tampil();
break;

case 5:
listDosen.insertionSort();
System.out.println(x:"\nData setelah Insertion Sort ASCENDING:");
listDosen.tampil();
break;

case 6:
System.out.print(s:"\nMasukkan nama dosen yang dicari: ");
String cariNama = sc.nextLine();
listDosen.sequentialSearchByNama(cariNama);
break;

case 7:
System.out.print(s:"\nMasukkan usia dosen yang dicari: ");
int cariUsia = sc.nextInt();
sc.nextLine();
listDosen.binarySearchByUsia(cariUsia);
break;

case 8:

```

```

case 8:
System.out.println(x:"Program selesai.");
System.exit(status:0);
break;

default:
System.out.println(x:"Pilihan tidak valid, coba lagi.");
}

```

```

}

```

```

public class DataDosen12 {
    Dosen12[] dataDosen = new Dosen12[10];
    int idx = 0;

    public void tambah(Dosen12 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println(x:"Data dosen sudah penuh!");
        }
    }

    public void tampil() {
        if (idx == 0) {
            System.out.println(x:"Tidak ada data dosen.");
        } else {
            for (int i = 0; i < idx; i++) {
                dataDosen[i].tampil();
            }
        }
    }

    public void SortingASC() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - 1 - i; j++) {
                if (dataDosen[j].usia > dataDosen[j + 1].usia) {
                    Dosen12 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
        System.out.println(x:"Data telah diurutkan secara ASCENDING berdasarkan usia.");
    }

    public void SortingDSC() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - 1 - i; j++) {
                if (dataDosen[j].usia < dataDosen[j + 1].usia) {
                    Dosen12 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
    }
}

```

```

public void insertionSort() {
    for (int i = 1; i < idx; i++) {
        Dosen12 temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia > temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
    System.out.println(x:"Data telah diurutkan dengan Insertion Sort (ASCENDING).");
}

public void sequentialSearchByNama(String nama) {
    int hasil = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(nama)) {
            System.out.println("\nData ditemukan pada indeks " + i + ":");
            dataDosen[i].tampil();
            hasil++;
        }
    }
    if (hasil == 0) {
        System.out.println("Data dengan nama '" + nama + "' tidak ditemukan.");
    } else if (hasil > 1) {
        System.out.println("Peringatan: Ditemukan " + hasil + " data dengan nama yang sama!");
    }
}

public void binarySearchByUsia(int usia) {
    SortingASC();

    int left = 0, right = idx - 1, hasil = 0;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (dataDosen[mid].usia == usia) {
            int i = mid;
            while (i >= 0 && dataDosen[i].usia == usia) {
                System.out.println("\nData ditemukan pada indeks " + i + ":");
                dataDosen[i].tampil();
                hasil++;
                i--;
            }
            i = mid + 1;
            while (i < idx && dataDosen[i].usia == usia) {
                System.out.println("\nData ditemukan pada indeks " + i + ":");
                dataDosen[i].tampil();
                hasil++;
                i++;
            }
            break;
        } else if (dataDosen[mid].usia < usia) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }
    if (hasil == 0) {
        System.out.println("Data dengan usia " + usia + " tidak ditemukan.");
    } else if (hasil > 1) {
        System.out.println("Peringatan: Ditemukan " + hasil + " data dengan usia yang sama!");
    }
}

```