

JOBSHEET VII SEARCHING

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. Searching / Pencarian Menggunakan Algoritma Sequential Search

Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa.

Mahasiswa
nim: String nama: String kelas: String ipk: double
Mahasiswa() Mahasiswa(nm: String, name: String, kls: String, ip: double) tampilInformasi(): void

Berdasarkan class diagram di atas, akan dibuat class Mahasiswa yang berfungsi untuk membuat objek mahasiswa yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga fungsi tampilInformasi() untuk menampilkan semua attribute yang ada.

MahasiswaBerprestasi
listMhs: Mahasiswa[5] idx: int
tambah(mhs: Mahasiswa): void tampil(): void sequentialSearch(double cari): int tampilPosisi(double x,int pos): void tampilDataSearch(double x,int pos) :void

Selanjutnya class diagram di atas merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array mahasiswa, misalkan untuk menambahkan objek mahasiswa, menampilkan semua data mahasiswa, untuk melakukan pencarian berdasarkan IPK menggunakan algoritma Sequential Search, menampilkan posisi dari data yang dicari, serta menampilkan data mahasiswa yang dicari.

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 **tampilPosisi** bertipe void dan Deklarasikan isi dari method **tampilPosisi** pada class **MahasiswaBerprestasi**<no presensi>.

```
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");
    }
}
```

5. Pada class **MahasiswaBerprestasi**<no presensi>, 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**<noPresensi> , tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.

```
public class MahasiswaDemo {
    Run | Debug
    public static void main(String[] args) {
        MahasiswaBerprestasi list = new MahasiswaBerprestasi();
        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(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 Mahasiswa(nim, nama, kelas, ipk));
        }

        list.tampil();
        //melakukan pencarian data sequential
        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:"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.

7.2.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini.

```
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
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
    
```

7.2.3. Pertanyaan

1. Jelaskan perbedaan metod `tampilDataSearch` dan `tampilPosisi` pada class MahasiswaBerprestasi!
2. Jelaskan fungsi `break` pada kode program dibawah ini!

```

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

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**. Kemudia panggil method **tampilPosisi** dan **tampilDataSearch**

```
//melakukan pencarian data 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 cari = sc.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

Cocokkan hasil kode program anda dengan gambar berikut ini.

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
-----
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
    
```

7.3.3. Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
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
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.



7.5. 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:
 - a. **PencarianDataSequential**<no presensi> : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.
 - b. **PencarianDataBinary**<no presensi> : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.
 - c. Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!