

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 7



ATHAULLA HAFIZH

244107020030

TI 1 E

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

Searching / Pencarian Menggunakan Algoritma Sequential Search

Kode Program Class MahasiswaBerprestasi05

```
package Jobsheet7;

public class MahasiswaBerprestasi05 {

    Mahasiswa05 [] listMhs = new Mahasiswa05[5];
    int idx;

    void tambah (Mahasiswa05 m) {
        if (idx < listMhs.length) {
            listMhs [idx] = m;
            idx++;
        } else {
            System.out.println ("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa05 m:listMhs) {
            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) {
                    Mahasiswa05 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;
            }
        }
        Mahasiswa05 tmp = listMhs[idxMin];
        listMhs [idxMin] = listMhs[i];
        listMhs [i] = tmp;
    }
}

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa05 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 MahasiswaDemo05

```

package Jobsheet7;

import java.util.Scanner;

public class MahasiswaDemo05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi05 list = new MahasiswaBerprestasi05();
        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.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 Mahasiswa05(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);
    }
}

```

Output

```
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
-----
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
```

Pertanyaan

1. Jelaskan perbedaan metod 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 dibawah 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 variabel 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 MahasiswaBerprestasi05

```
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;
}

```

Penambahan Kode Program Class MahasiswaDemo05

```

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);

```

Output

```

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

```

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

```

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). Untuk 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 MahasiswaDemo05

```
System.out.print(s:"Masukkan jumlah mahasiswa : ");
int jumMhs = sc.nextInt();
MahasiswaBerprestasi05 list = new MahasiswaBerprestasi05(jumMhs);
```

Penambahan Kode Program Class MahasiswaBerprestasi05

```
public class MahasiswaBerprestasi05 {
    Mahasiswa05 [] listMhs;
    int idx = 0;

    public MahasiswaBerprestasi05 (int jumMhs) {
        listMhs = new Mahasiswa05[jumMhs];
    }

    void tampil() {
        for (Mahasiswa05 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:
 - a. PencarianDataSequential : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.
 - b. PencarianDataBinary : 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!

Penambahan Kode Program Class DosenDemo05

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

DataDosen05 dosen = new DataDosen05(jumlah);
```

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

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

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

Penambahan Kode Program Class DataDosen05

```
Dosen05[] dataDosen;
int idx;

public DataDosen05(int jumlah) {
    dataDosen = new Dosen05[jumlah];
}
```

```
void pencarianDataSequential(String cariNama) {
    int jumlahHasil = 0;
    for (int i = 0; i < dataDosen.length; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(cariNama)) {
            System.out.println("Data dosen dengan nama : " + dataDosen[i].nama + " ditemukan pada indeks " + i);
            System.out.println(x:"Dengan detail data dosen : ");
            System.out.println("Kode      : " + dataDosen[i].kode);
            System.out.println("Nama       : " + dataDosen[i].nama);
            if (dataDosen[i].jenisKelamin == true) {
                System.out.println(x:"Jenis Kelamin : Wanita");
            } else {
                System.out.println(x:"Jenis Kelamin : Pria");
            }
            System.out.println("Usia       : " + dataDosen[i].usia);
            System.out.println();
            jumlahHasil++;
        }
    }
    if (jumlahHasil == 0) {
        System.out.println(x:"Data tidak ditemukan.");
    } else {
        System.out.println(x:"=====");
        System.out.println("PERINGATAN : Data ditemukan sebanyak " + jumlahHasil);
        System.out.println(x:"=====");
        if (jumlahHasil > 1) {
            System.out.println(x:"PERINGATAN: Terdapat lebih dari satu data dosen dengan nama tersebut.");
        }
    }
}
```

```

void pencarianDataBinary(int cari, int left, int right) {
    int mid;
    int jumlahHasil = 0;
    int[] indexSama = new int[dataDosen.length];

    while (right >= left) {
        mid = (left + right) / 2;

        if (cari == dataDosen[mid].usia) {
            indexSama[jumlahHasil] = mid;
            jumlahHasil++;

            int temp = mid - 1;
            while (temp >= left && dataDosen[temp].usia == cari) {
                indexSama[jumlahHasil] = temp;
                jumlahHasil++;
                temp--;
            }

            temp = mid + 1;
            while (temp <= right && dataDosen[temp].usia == cari) {
                indexSama[jumlahHasil] = temp;
                jumlahHasil++;
                temp++;
            }

            System.out.println(x:"=====");
            System.out.println("PERINGATAN : Data ditemukan sebanyak " + jumlahHasil);
            System.out.println(x:"=====");
            for (int i = 0; i < jumlahHasil; i++) {
                int index = indexSama[i];
                System.out.println("Data dosen dengan usia : " + dataDosen[index].usia + " ditemukan pada indeks " + index);
                System.out.println(x:"Dengan detail data dosen : ");
                System.out.println("Kode          : " + dataDosen[index].kode);
                System.out.println("Nama           : " + dataDosen[index].nama);
                if (dataDosen[index].jenisKelamin == true) {
                    System.out.println(x:"Jenis Kelamin : Wanita");
                } else {
                    System.out.println(x:"Jenis Kelamin : Pria");
                }
                System.out.println("Usia           : " + dataDosen[index].usia);
                System.out.println();
            }

            if (jumlahHasil > 1) {
                System.out.println(x:"PERINGATAN: Terdapat lebih dari satu data dosen dengan usia tersebut.");
            }
            return;
        } else if (dataDosen[mid].usia < cari) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }
    System.out.println(x:"Data tidak ditemukan.");
}

```

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.