

LAPORAN HASIL PRATIKUM
ANALISIS DAN STRUKTUR DATA
JOBSHEET 7



Raihan Akbar Putra Prasetyo/244107020087

Kelas: TI-1E

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
PRAKTIKUM 25

Percobaan Sequential Search

- **CODE PROGRAM**

- 1. Penambahan di dalam class mahasiswaBerprestasi24.java**

```
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");
    }
}
```

- 2. Perubahan di mahasiswaDemo24.java**

```
list.tampil();
//melakukan pencarian data sequential
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.sequentialSearching(cari);
int pss = (int)posisi;
list.tampilPosisi(cari, pss);
list.tampilDataSearch(cari, pss);
```

- **Output**

```
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
PS D:\kuliah\PRAKTIKUM-ASD>
```

- **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;
}
```

1. Method `tampilPosisi()` digunakan untuk menampilkan apakah data mahasiswa dengan IPK tertentu ditemukan dan berada di indeks ke berapa, sedangkan `tampilDataSearch()` menampilkan detail lengkap mahasiswa (NIM, nama, kelas, IPK) berdasarkan indeks tersebut.
2. Untuk menghentikan loop saat posisi indeks sudah ditemukan

Searching / Pencarian Menggunakan Binary Search

- **Code program**

- 1. Perubahan pada mahasiswaberprestasi**

```
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. Perubahan pada mahasiswademo**

```
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("-----
-----");
    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**

```
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
PS D:\kuliah\PRAKTIKUM-ASD>
```

- **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!**

Program tetep bisa berjalan tapi hasilnya mungkin bisa salah karena sifatnya harus urut

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

```
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;
}
```

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

```
package jobsheet7;
import java.util.Scanner;
public class MahasiswaDemo24 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        MahasiswaBerprestasi24 list = new MahasiswaBerprestasi24();

        System.out.print("Masukkan jumlah mahasiswa: ");
        int jumlah = input.nextInt();
        input.nextLine();

        for (int i = 0; i < jumlah; 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: ");
            double ipk = input.nextDouble();
            input.nextLine();

            Mahasiswa24 m = new Mahasiswa24(nim, nama, kelas, ipk);
            list.tambah(m);
        }

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

Latihan

- **Code program**
 - a. **Penambahan method**

```
void pencarianDataSequential13(String cariNama){
    int cari = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(cariNama)) {
            dataDosen[i].tampil();
            cari++;
        }
    }
    if (cari == 0) {
        System.out.println("dosen dengan nama " + cariNama + " tidak ditemukan");
    } else if (cari > 1) {
        System.out.println("ditemukan lebih dari satu data dosen dengan nama yang sama!");
    }
}

void pencarianDataBinary(int cariUsia){
    int awal = 0, akhir = idx - 1, tengah = -1;
    boolean ditemukan = false;

    while (awal <= akhir) {
        tengah = (awal + akhir) / 2;
        if (dataDosen[tengah].usia == cariUsia) {
            ditemukan = true;
            break;
        }
        if (dataDosen[tengah].usia < cariUsia) {
            awal = tengah + 1;
        } else {
            akhir = tengah - 1;
        }
    }
    if (!ditemukan) {
        System.out.println("Dosen dengan usia " + cariUsia + " tidak ditemukan");
        return;
    }

    int kiri = tengah, kanan = tengah;
    while (kiri >= 0 && dataDosen[kiri].usia == cariUsia) kiri--;
    while (kanan < idx && dataDosen[kanan].usia == cariUsia) kanan++;
    System.out.println("hasil pencarian dosen dengan usia " + cariUsia + ": ");

    for (int i = kiri + 1; i < kanan; i++) {
        dataDosen[i].tampil();
    }
    if (kanan - kiri - 1 > 1) {
        System.out.println("Peringatan! ada lebih dari 1 dosen dengan usia yang sama");
    }
}

}
```


b. Dosenmain

```
package jobsheet7;
import java.util.Scanner;
public class DosenMain24 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen24 listDosen = new DataDosen24();

        while (true) {
            System.out.println("===== Menu =====");
            System.out.println("1. Tambahkan Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Urutkan Data Bubble Sort (ASC)");
            System.out.println("4. Urutkan Data Selection Sort (DSC)");
            System.out.println("5. Cari Data dengan Insertion Sort + Binary
Search");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Kode: ");
                    String kode = sc.nextLine();
                    System.out.print("Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("Jenis Kelamin (L = true, P = false): ");
                    boolean jk = sc.nextBoolean();
                    System.out.print("Usia: ");
                    int usia = sc.nextInt();
                    sc.nextLine();
                    listDosen.tambah(new Dosen24(kode, nama, jk, usia));
                    break;
                case 2:
                    listDosen.tampil();
                    break;
                case 3:
                    listDosen.sortingASC();
                    System.out.println("Data setelah sorting ASC:");
                    listDosen.tampil();
                    break;
                case 4:
                    listDosen.sortingDSC();
                    System.out.println("Data setelah sorting DSC:");
                    listDosen.tampil();
                    break;
                case 5:
                    System.out.print("Masukkan usia dosen yang ingin dicari: ");
                    int usiaCari = sc.nextInt();
                    sc.nextLine();
                    listDosen.sortingASC();
                    listDosen.pencarianDataBinary(usiaCari);
                    break;
                case 0:
                    System.out.println("Keluar dari program.");
                    sc.close();
                    return;
                default:
                    System.out.println("Pilihan tidak valid!");
            }
        }
    }
}
```

