

Jobsheet 7 – Bisma Adhiakas – Searching

Percobaan 1

1. Buat folder dengan nama jobsheet7 lalu copy class Mahasiswa07, MahasiswaBerprestasi07, dan MahasiswaDemo07 dari folder jobsheet6 ke folder jobsheet7
2. Pada class MahasiswaBerprestasi07 tambah method algoritma sequential search.

```
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;  
}
```

3. Lalu tambah method untuk menampilkan posisi pada class MahasiswaBerprestasi07

```
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");  
    }  
}
```

4. Masih pada class yang sama, tambahkan 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);  
    }  
}
```

5. Pada class MahasiswaDemo07 tambahkan inisialisasi int jumMhs dan ubah kondisi pada for untuk memasukkan data, lalu tambahkan tampilan untuk pemanggilan method tampil

```
Run | Debug | Run main | Debug main
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    MahasiswaBerprestasi07 list = new MahasiswaBerprestasi07();
    Mahasiswa07 Mhs[] = new Mahasiswa07[list.listMhs.length];
    int jumMhs = 5;
    for (int i = 0; i < jumMhs; i++) {
        if (i >= Mhs.length) {
            System.out.println(x:"Data sudah penuh");
        } else {
            System.out.println("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 ipk = sc.nextLine();
            Double ipk = Double.parseDouble(ipk);
            System.out.println(x:"-----");
            list.tambah(new Mahasiswa07(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);
}
```

6. Jalankan program yang sudah dibuat tadi

```
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 : 3.3
-----
```

```
Masukkan Data Mahasiswa ke-5
NIM : 555
Nama : fia
Kelas : 2
IPK : 3.5
-----
```

```
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: 3.3
-----
```

```
Nama: fia
NIM: 555
Kelas: 2
IPK: 3.5
-----
```

```
-----
Pencarian Data
-----
```

```
Masukkan IPK Mahasiswa yang Dicari:
IPK: 3.5
Menggunakan Sequential Searching
data mahasiswa dengan IPK :3.5 ditemukan pada indeks 4
NIM      : 555
Nama     : fia
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;  
}
```

Jawaban

1. TampilDataSearch adalah method untuk menampilkan nilai dari data pada posisi ditemukannya nilai ipk yang sama, sedangkan tampilPosisi digunakan hanya untuk menampilkan pada indeks nilai dari ipk yang sama ditemukan
2. Untuk mengecek apakah nilai ipk pada array listMhs indeks j sama dengan nilai dari cari, lalu jika sama, posisi akan menyalin nilai dari j dan akan langsung break

Percobaan 2

1. Pada class MahasiswaBerprestasi07 tambahkan method findBinarySearch

```
int findBinarySeach(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 findBinarySeach(cari, left, mid-1);  
        }else{  
            return findBinarySeach(cari, mid+1, right);  
        }  
    }  
    return -1;  
}
```

2. Pada class MahasiswaDemo07 panggil method findBinarySearch, kemudian panggil method tampilPosisi dan tampilDataSearch

```
//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: ");
String cariIPK = sc.nextLine();
double cari = Double.parseDouble(cariIPK);

System.out.println(x:"Menggunakan Sequential Searching");
double posisi2 = list.findBinarySeach(cari, left:0, jumMhs-1);
int pss2 = (int)posisi2;
list.tampilPosisi(cari, pss2);
list.tampilDataSearch(cari, pss2);
```

3. Jalankan Program dengan input urut secara ascending

```
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 Sequential Searching  
data mahasiswa dengan IPK :3.7 ditemukan pada indeks 4  
NIM : 555  
Nama : anita  
Kelas : 2  
IPK : 3.7
```

```
PS D:\Kuliah\Semester2\PRAKTIKUM-ASD\jobsheet7>
```

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

Jawaban

1.

```
mid = (left + right) / 2;
if (cari == listMhs[mid].ipk) {
    return (mid);
}else if(listMhs[mid].ipk>cari){
    return findBinarySeach(cari, left, mid-1);
}else{
    return findBinarySeach(cari, mid+1, right);
}
```
- 2.
3. Bisa berjalan, tetapi data yang dicari bisa berkemungkinan tidak ditemukan. Karena pengecekan tidak satu per satu, secara urut, tetapi mengecek nilai apakah lebih besar atau tidak sehingga urutan nilai penting.

4. Tidak sesuai, karena jika dicek, posisi right akan dilanjut ke mid -1

```
int findBinarySeach(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 findBinarySeach(cari, left, mid-1);
        } else {
            return findBinarySeach(cari, mid+1, right);
        }
    }
    return -1;
}
```

Nama: 2
NIM: 1
Kelas: 3
IPK: 3.8

Nama: 5
NIM: 4
Kelas: 6
IPK: 3.7

Nama: 8
NIM: 7
Kelas: 9
IPK: 3.5

Nama: q
NIM: 1
Kelas: 2
IPK: 3.4

Nama: w
NIM: 2
Kelas: 2
IPK: 3.2

Pencarian Data

Masukkan IPK Mahasiswa yang Dicari:
IPK: 3.2
Menggunakan Sequential Searching
data mahasiswa dengan IPK :3.2 ditemukan pada indeks 4
NIM : 2
Nama : w
Kelas : 2
IPK : 3.2


```

public class MahasiswaBerprestasi07 {

    int jumMhs;
    Mahasiswa07 listMhs[] = new Mahasiswa07[jumMhs];
    int idx;

    public MahasiswaBerprestasi07(int jumlah){
        this.jumMhs = jumlah;
        listMhs = new Mahasiswa07[jumlah];
    }
}

```

5.

```

import java.util.Scanner;

public class MahasiswaDemo07 {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Masukkan jumlah mahasiswa yang ingin didata: ");
        int jumMhs = sc.nextInt();
        sc.nextLine();
        MahasiswaBerprestasi07 list = new MahasiswaBerprestasi07(jumMhs);
        list.jumMhs = jumMhs;
        Mahasiswa07 Mhs[] = new Mahasiswa07[jumMhs];
    }
}

```

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!

```
1
2 public class Dosen {
3
4     String kode, nama;
5     boolean jenisKelamin; // true = laki, false perempuan
6     int usia;
7
8     Dosen(String kd, String name, boolean jk, int age) {
9         kode = kd;
10        nama = name;
11        jenisKelamin = jk;
12        usia = age;
13    }
14    Dosen(){
15
16    }
17
18    void tampilInfoDosen() {
19        System.out.println("Kode: " + kode);
20        System.out.println("Nama: " + nama);
21        if (jenisKelamin) {
22            System.out.println(x:"Jenis Kelamin: Pria");
23        } else {
24            System.out.println(x:"Jenis Kelamin: Wanita");
25        }
26        System.out.println("Usia: " + usia);
27    }
28 }
29
```

```

public class DataDosen {

    Dosen dataDosen[] = new Dosen[10];
    int idx;

    void tambahDosen(Dosen dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println(x:"data sudah penuh");
        }
    }

    void tampil() {
        for (Dosen dsn : dataDosen) {
            if (dsn != null) {
                dsn.tampilInfoDosen();
                System.out.println(x:"-----");
            }
        }
    }

    void SortingASC() {
        for (int i = 0; i < dataDosen.length; i++) {
            for (int j = 1; j < dataDosen.length - i; j++) {
                if (dataDosen[j].usia > dataDosen[j - 1].usia) {
                    Dosen temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j - 1];
                    dataDosen[j - 1] = temp;
                }
            }
        }
    }

    void SortingOSC() {
        for (int i = 0; i < dataDosen.length - 1; i++) {
            int idxMax = i;
            for (int j = i + 1; j < dataDosen.length; j++) {
                if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                    idxMax = j;
                }
            }
            Dosen tmp = dataDosen[idxMax];
            dataDosen[idxMax] = dataDosen[i];
            dataDosen[i] = tmp;
        }
    }
}

```

```

void InsertionSort() {
    for (int i = 1; i < dataDosen.length; i++) {
        Dosen temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
}

int PencarianDataSequential(String cari) {
    int posisi = -1;
    int ditemukan = 0;
    for (int j = 0; j < dataDosen.length; j++) {
        if (dataDosen[j].nama.equalsIgnoreCase(cari)) {
            ditemukan++;
            if (ditemukan == 1) {
                posisi = j;
            } else if (ditemukan >= 2) {
                System.out.println(x: "\nDATA DITEMUKAN LEBIH DARI 1!");
            }
        }
    }
    return posisi;
}

void tampilPosisi(String x, int pos) {
    if (pos != -1) {
        System.out.println("data mahasiswa dengan usia : " + x + " ditemukan pada indeks " + pos);
    } else {
        System.out.println("data " + x + " tidak ditemukan");
    }
}

void tampilDataSearch07(int pos) {
    if (pos != -1) {
        System.out.println("Kode\t: " + dataDosen[pos].kode);
        System.out.println("Nama\t: " + dataDosen[pos].nama);
        if (dataDosen[pos].jenisKelamin) {
            System.out.println(x: "Jenis Kelamin: Pria");
        } else {
            System.out.println(x: "Jenis Kelamin: Wanita");
        }
        System.out.println("Usia\t: " + dataDosen[pos].usia);
    }
}

```

```

//DSC sort dulu
int PencarianDataBinary07(int ditemukan, int cari, int left, int right) {
    int mid;
    int posisi = -1;
    if (right >= left) {
        mid = (left + right) / 2;
        if (cari == dataDosen[mid].usia) {
            ditemukan++;
            posisi = mid;
        } else if (dataDosen[mid].usia < cari || mid != 0) {
            return PencarianDataBinary07(ditemukan, cari, left, mid - 1);
        } else if (dataDosen[mid].usia > cari || mid != dataDosen.length){
            return PencarianDataBinary07(ditemukan, cari, mid + 1, right);
        }
    }

    return posisi;
}

```

```

2  import java.util.Scanner;
3
4  public class DosenMain {
5
6      Run main | Debug main | Run | Debug
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9          DataDosen list = new DataDosen();
10         Dosen dsn[] = new Dosen[list.dataDosen.length];
11         boolean pilihan = true;
12         int choose;
13         String gender;
14
15         while (pilihan) {
16             System.out.println(x:"MENU DOSEN");
17             System.out.println(x:"1. Tambah Data Dosen");
18             System.out.println(x:"2. Tampil Informasi Seluruh Dosen");
19             System.out.println(x:"3. Sorting Ascending");
20             System.out.println(x:"4. Sorting Desscending");
21             System.out.println(x:"5. Cari Berdasarkan Nama Dosen");
22             System.out.println(x:"6. Cari Berdasarkan Usia");
23             System.out.println(x:"0. Keluar");
24             System.out.print(s:"Pilihan: ");
25             choose = sc.nextInt();
26             sc.nextLine();
27             switch (choose) {
28                 case 0:
29                     pilihan = false;
30                     break;
31                 case 1:
32                     for (int i = 0; i < dsn.length; i++) {
33                         if (i >= dsn.length) {
34                             System.out.println(x:"Data sudah penuh");
35                         } else {
36                             System.out.println(x:"-----");
37                             System.out.println("Dosen ke-" + (i + 1));
38                             dsn[i] = new Dosen();
39                             System.out.print(s:"Masukkan Kode Dosen: ");
40                             dsn[i].kode = sc.nextLine();
41                             System.out.print(s:"Nama Dosen: ");
42                             dsn[i].nama = sc.nextLine();
43                             System.out.print(s:"Masukkan Jenis Kelamin(Pria/Wanita): ");
44                             gender = sc.nextLine();
45                             if (gender.equalsIgnoreCase(anotherString:"Pria")) {
46                                 dsn[i].jenisKelamin = true;
47                             } else if (gender.equalsIgnoreCase(anotherString:"Wanita")) {
48                                 dsn[i].jenisKelamin = false;
49                             } else {
50                                 System.out.println(x:"Gender salah, harp mengulang");
51                                 continue;
52                             }
53                             System.out.print(s:"Masukkan Usia: ");
54                             dsn[i].usia = sc.nextInt();
55                             sc.nextLine();
56                             list.tambahDosen(dsn[i]);
57                         }
58                     }
59                     break;

```

```

59         case 2:
60             list.tampil();
61             break;
62         case 3:
63             list.SortingASC();
64             list.tampil();
65             break;
66         case 4:
67             list.SortingDESC();
68             list.tampil();
69             break;
70         case 5:
71             System.out.print(s:"Masukkan nama dosen: ");
72             String cari = sc.nextLine();
73             int posisi = list.PencarianDataSequential(cari);
74             list.tampilDataSearch07(posisi);
75             break;
76         case 6:
77             System.out.print(s:"Masukkan usia yang ingin dicari: ");
78             int age = sc.nextInt();
79             list.SortingDESC();
80             int posisi2 = list.PencarianDataBinary07(ditemukan:0, age, left:0, dsn.length-1);
81             list.tampilDataSearch07(posisi2);
82             break;
83         default:
84             break;
85     }
86 }
87 }
88 }
89

```