

## LAPORAN JOBSHEET 7

### PERCOBAAN 1

#### Class mahasiswa9

```
package JOBSHEET7;

import java.util.Scanner;

public class Mahasiswa9 {
    String nim, nama, kelas;
    double ipk;

    Mahasiswa9() {
    }

    Mahasiswa9(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilInformasi() {
        System.out.println("Nama: " + nama);
        System.out.println("NIM: " + nim);
        System.out.println("Kelas: " + kelas);
        System.out.println("IPK: " + ipk);
    }
}
```

## Class MahasiswaBerprestasi9

```
package JOBSHEET7;

public class MahasiswaBerprestasi9 {
    Mahasiswa9[] listMhs = new Mahasiswa9[5];
    int idx;

    void tambah(Mahasiswa9 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println(x:"Data Sudah Penuh");
        }
    }

    void tampil() {
        for (Mahasiswa9 m : listMhs) {
            if (m != null) {
                m.tampilInformasi();
                System.out.println(x:"-----");
            }
        }
    }

    int sequentialSearching(double cari) {
        int posisi = -1;
        for (int j = 0; j < listMhs.length; j++) {
            if (listMhs[j] != null && 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 dengan IPK " + x + " tidak ditemukan");
        }
    }

    void tampilDataSearch(double x, int pos) {
        if (pos != -1) {
            System.out.println("NIM      : " + listMhs[pos].nim);
            System.out.println("Nama    : " + listMhs[pos].nama);
            System.out.println("Kelas  : " + listMhs[pos].kelas);
            System.out.println("IPK     : " + x);
        } else {
            System.out.println("Data mahasiswa dengan IPK " + x + " tidak ditemukan");
        }
    }
}
```

## Class MahasiswaDemo9

```
1 package JOBSHEET7;
2
3 import java.util.Scanner;
4
5 public class MahasiswaDemo9 {
6     Run | Debug
7     public static void main(String[] args) {
8         MahasiswaBerprestasi9 list = new MahasiswaBerprestasi9();
9         Scanner sc = new Scanner(System.in);
10        int jumMhs = 5;
11
12        for (int i = 0; i < jumMhs; i++) {
13            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));
14            System.out.print(s:"NIM    : ");
15            String nim = sc.nextLine();
16            System.out.print(s:"Nama  : ");
17            String nama = sc.nextLine();
18            System.out.print(s:"Kelas : ");
19            String kelas = sc.nextLine();
20            System.out.print(s:"IPK   : ");
21            double ipk = sc.nextDouble();
22            sc.nextLine();
23
24            System.out.println(x:"-----");
25            list.tambah(new Mahasiswa9(nim, nama, kelas, ipk));
26        }
27
28        list.tampil();
29
30        System.out.println(x:"-----");
31        System.out.println(x:"Pencarian data");
32        System.out.println(x:"-----");
33        System.out.println(x:"Masukkan IPK mahasiswa yang dicari: ");
34        System.out.print(s:"IPK : ");
35        double cari = sc.nextDouble();
36
37        System.out.println(x:"Menggunakan Sequential searching");
38        double posisi = list.sequentialSearching(cari);
39        int pss = (int) posisi;
40        list.tampilPosisi(cari, pss);
41        list.tampilDataSearch(cari, pss);
42    }
43 }
```

## Run program

```
Nama: adi
NIM: 1111
Kelas: 2
IPK: 3.6
-----
Nama: to
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!
2. Jelaskan fungsi `break` pada kode program dibawah ini!

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

## Jawaban

1. `tampilPosisi` Berfungsi untuk menampilkan posisi (indeks) di mana data mahasiswa dengan IPK tertentu ditemukan di array. Sedangkan `tampilDataSearch` Berfungsi untuk menampilkan informasi lengkap mahasiswa (NIM, nama, kelas, dan IPK) yang ditemukan berdasarkan IPK yang dipanggil.
2. `break` berfungsi menghentikan perulangan `for` pada saat ditemukan IPK yang cocok dengan nilai cari.

## PERCOBAAN 2

### Class MahasiswaBerprestasi

```
void bubbleSort() {
    for (int i = 0; i < listMhs.length - 1; i++) {
        for (int j = 0; j < listMhs.length - 1 - i; j++) {
            if (listMhs[j].ipk < listMhs[j + 1].ipk) {
                Mahasiswa9 tmp = listMhs[j];
                listMhs[j] = listMhs[j + 1];
                listMhs[j + 1] = tmp;
            }
        }
    }
}

int findBinarySearch(double cari, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;
        if (listMhs[mid].ipk == cari) {
            return mid;
        } else if (listMhs[mid].ipk > cari) {
            return findBinarySearch(cari, left, mid - 1);
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}
```

## Class MahasiswaDemo

```
public class MahasiswaDemo9 {  
    Run | Debug  
    public static void main(String[] args) {  
        MahasiswaBerprestasi9 list = new MahasiswaBerprestasi9();  
        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    : ");  
            double ipk = sc.nextDouble();  
            sc.nextLine();  
  
            System.out.println(x:"-----");  
            list.tambah(new Mahasiswa9(nim, nama, kelas, ipk));  
        }  
  
        list.tampil();  
        list.bubbleSort();  
  
        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 Binary Search");  
        double posisi = list.findBinarySearch(cari, left:0, jumMhs-1);  
        int pss = (int) posisi;  
        list.tampilPosisi(cari, pss);  
        list.tampilDataSearch(cari, pss);  
    }  
}
```

## Run Program

```
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 searching
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!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data IPK yang dimasukkan tidakurut. 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.

## Jawaban

1. 

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

```

if (right >= left) {
    int mid = (left + right) / 2;
    if (listMhs[mid].ipk == cari) {
        return mid;
    } else if (listMhs[mid].ipk > cari) {
        return findBinarySearch(cari, left, mid - 1);
    } else {
        return findBinarySearch(cari, mid + 1, right);
    }
}

```

- 2.
3. Tidak, Binary search mengasumsikan bahwa data sudah terurut (baik naik atau turun). Kalau tidak, hasil pencariannya tidak bisa dijamin benar.
4. Data yang diurutkan dari besar ke kecil tidak akan cocok dengan binary search biasa (yang asumsinya kecil ke besar), sehingga hasil pencarian tidak akan sesuai.

```

int findBinarySearch(double cari, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;
        if (listMhs[mid].ipk == cari) {
            return mid;
        } else if (listMhs[mid].ipk < cari) {
            return findBinarySearch(cari, left, mid - 1);
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}

```

- 5.

```

package JOBSHEET7;

import java.util.Scanner;

public class MahasiswaDemo9 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

        MahasiswaBerprestasi9 list = new MahasiswaBerprestasi9(jumMhs);
    }
}

```



```

package JOBSHEET7;

public class MahasiswaBerprestasi9 {
    Mahasiswa9[] listMhs;
    int idx;

    MahasiswaBerprestasi9(int jumlah) {
        listMhs = new Mahasiswa9[jumlah];
        idx = 0;
    }
}

```

### 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!

## Class dosen

```
package JOBSHEET7;

public class Dosen {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    public Dosen(String kode, String nama, boolean jenisKelamin, int usia) {
        this.kode = kode;
        this.nama = nama;
        this.jenisKelamin = jenisKelamin;
        this.usia = usia;
    }

    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);
    }
}
```

## Class datadosen

```
package JOBSHEET7;

public class DataDosen {
    Dosen[] dataDosen = new Dosen[10];
    int idx = 0;

    public void tambah(Dosen dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        }
    }

    public void tampil() {
        if (idx == 0) {
            System.out.println("Data dosen kosong."++);
            return;
        }

        for (int i = 0; i < idx; i++) {
            System.out.println("Data Dosen ke-" + (i + 1) + ":");
            dataDosen[i].tampil();
            System.out.println();
        }
    }

    public void sortingASC() {
```

```

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

    public void sortingDSC() {

        for (int i = 1; i < idx; i++) {
            Dosen temp = dataDosen[i];
            int j = i - 1;

            while (j >= 0 && dataDosen[j].usia < temp.usia) {
                dataDosen[j + 1] = dataDosen[j];
                j = j - 1;
            }
            dataDosen[j + 1] = temp;
        }
        tampil();
    }

    public void pencarianDataSequential(String namaCari) {
        int hasilPencarian = 0;
        for (int i = 0; i < idx; i++) {
            if (dataDosen[i].nama.equalsIgnoreCase(namaCari)) {
                System.out.println("Data Dosen ditemukan pada indeks ke-" + i
+ ":");

                dataDosen[i].tampil();
                hasilPencarian++;
            }
        }
        if (hasilPencarian == 0) {
            System.out.println("Data Dosen dengan nama '" + namaCari + "'
tidak ditemukan.");
        } else if (hasilPencarian > 1) {
            System.out.println("Peringatan! Ditemukan lebih dari satu data
dosen dengan nama '" + namaCari + "'.");
        }
    }

    public void pencarianDataBinary(int usiaCari) {
        sortingASC();
    }

```

```

int low = 0;
int high = idx - 1;
int hasilPencarian = 0;
int[] indeksDitemukan = new int[idx];
int jumlahDitemukan = 0;

while (low <= high) {
    int mid = low + (high - low) / 2;

    if (dataDosen[mid].usia == usiaCari) {
        indeksDitemukan[jumlahDitemukan++] = mid;
        hasilPencarian++;

        int temp = mid - 1;
        while (temp >= 0 && dataDosen[temp].usia == usiaCari) {
            indeksDitemukan[jumlahDitemukan++] = temp;
            hasilPencarian++;
            temp--;
        }
        temp = mid + 1;
        while (temp < idx && dataDosen[temp].usia == usiaCari) {
            indeksDitemukan[jumlahDitemukan++] = temp;
            hasilPencarian++;
            temp++;
        }
        break;
    } else if (dataDosen[mid].usia < usiaCari) {
        low = mid + 1;
    } else {
        high = mid - 1;
    }
}

if (hasilPencarian == 0) {
    System.out.println("Data Dosen dengan usia " + usiaCari + " tidak ditemukan.");
} else if (hasilPencarian == 1) {
    System.out.println("Data Dosen ditemukan pada indeks ke-" + indeksDitemukan[0] + ":");
    dataDosen[indeksDitemukan[0]].tampil();
} else if (hasilPencarian > 1) {
    System.out.println("Peringatan! Ditemukan " + hasilPencarian + " data dosen dengan usia " + usiaCari + " pada indeks:");
    for (int i = 0; i < jumlahDitemukan; i++) {
        System.out.println("- " + indeksDitemukan[i]);
        dataDosen[indeksDitemukan[i]].tampil();
    }
}

```

```

        System.out.println();
    }
}
}
}

```

## Class DosenMain

```

package JOBSHEET7;

public class DataDosen {
    Dosen[] dataDosen = new Dosen[10];
    int idx = 0;

    public void tambah(Dosen dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        }
    }

    public void tampil() {
        if (idx == 0) {
            System.out.println("Data dosen kosong."++);
            return;
        }

        for (int i = 0; i < idx; i++) {
            System.out.println("Data Dosen ke-" + (i + 1) + ":");
            dataDosen[i].tampil();
            System.out.println();
        }
    }

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

```

```

public void sortingDSC() {

    for (int i = 1; i < idx; i++) {
        Dosen temp = dataDosen[i];
        int j = i - 1;

        while (j >= 0 && dataDosen[j].usia < temp.usia) {
            dataDosen[j + 1] = dataDosen[j];
            j = j - 1;
        }
        dataDosen[j + 1] = temp;
    }
    tampil();
}

public void pencarianDataSequential(String namaCari) {
    int hasilPencarian = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(namaCari)) {
            System.out.println("Data Dosen ditemukan pada indeks ke-" + i
+ ":");

            dataDosen[i].tampil();
            hasilPencarian++;
        }
    }
    if (hasilPencarian == 0) {
        System.out.println("Data Dosen dengan nama '" + namaCari + "'
tidak ditemukan.");
    } else if (hasilPencarian > 1) {
        System.out.println("Peringatan! Ditemukan lebih dari satu data
dosen dengan nama '" + namaCari + "'.");
    }
}

public void pencarianDataBinary(int usiaCari) {
    sortingASC();

    int low = 0;
    int high = idx - 1;
    int hasilPencarian = 0;
    int[] indeksDitemukan = new int[idx];
    int jumlahDitemukan = 0;

    while (low <= high) {
        int mid = low + (high - low) / 2;

        if (dataDosen[mid].usia == usiaCari) {
            indeksDitemukan[jumlahDitemukan++] = mid;
            hasilPencarian++;
        }
    }
}

```

```

        int temp = mid - 1;
        while (temp >= 0 && dataDosen[temp].usia == usiaCari) {
            indeksDitemukan[jumlahDitemukan++] = temp;
            hasilPencarian++;
            temp--;
        }
        temp = mid + 1;
        while (temp < idx && dataDosen[temp].usia == usiaCari) {
            indeksDitemukan[jumlahDitemukan++] = temp;
            hasilPencarian++;
            temp++;
        }
        break;
    } else if (dataDosen[mid].usia < usiaCari) {
        low = mid + 1;
    } else {
        high = mid - 1;
    }
}

if (hasilPencarian == 0) {
    System.out.println("Data Dosen dengan usia " + usiaCari + " tidak ditemukan.");
} else if (hasilPencarian == 1) {
    System.out.println("Data Dosen ditemukan pada indeks ke-" + indeksDitemukan[0] + ":");
    dataDosen[indeksDitemukan[0]].tampil();
} else if (hasilPencarian > 1) {
    System.out.println("Peringatan! Ditemukan " + hasilPencarian + " data dosen dengan usia " + usiaCari + " pada indeks:");
    for (int i = 0; i < jumlahDitemukan; i++) {
        System.out.println("- " + indeksDitemukan[i]);
        dataDosen[indeksDitemukan[i]].tampil();
        System.out.println();
    }
}
}
}
}

```

## Run Program

```
6. Pencarian Data (Binary - Usia)
8. Keluar
Pilihan: 5
Masukkan nama dosen yang dicari: jojo
Data Dosen ditemukan pada indeks ke-2:
Kode: 3
Nama: jojo
Jenis Kelamin: Laki-laki
Usia: 22
Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC Mengurutkan Dosen Termuda Hingga Tertua
4. Sorting DSC Mengurutkan Dosen Tertua Hingga Termuda
5. Pencarian Data (Sequential - Nama)
6. Pencarian Data (Binary - Usia)
8. Keluar
Pilihan: 6
Masukkan usia dosen yang dicari: 28
Data Dosen ke-1:
Kode: 1
Nama: koki
Jenis Kelamin: Laki-laki
Usia: 28

Data Dosen ke-2:
Kode: 2
Nama: koko
Jenis Kelamin: Laki-laki
Usia: 28

Data Dosen ke-3:
Kode: 3
Nama: jojo
Jenis Kelamin: Laki-laki
Usia: 22

Peringatan! Ditemukan 2 data dosen dengan usia 28 pada indeks:
- 1
Kode: 2
Nama: koko
Jenis Kelamin: Laki-laki
Usia: 28

- 8
Kode: 1
Nama: koki
Jenis Kelamin: Laki-laki
Usia: 28
```