

# **LAPORAN JOBSHEET 2**

## **KONSEP DASAR PEMROGRAMAN**

Mata Kuliah : Algoritma dan Struktur Data

Dosen : **Mungki Astiningrum, S.T., M.Kom.**



**Alfreda Dhaifullah Mahezwara**

**244107020219**

**Kelas : 1A**

**Absen : 04**

**PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN  
TEKNOLOGI INFORMASI POLITEKNIK NEGERI  
MALANG TAHUN 2025**

# Praktikum 1

```
package Pertemuan_7;

import java.util.Scanner;

public class Mahasiswa04 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    Mahasiswa04 () {

    }

    Mahasiswa04 (String nm, String nim, String kls, double ip) {

        nama = nm;

        this.nim = nim;

        kelas = kls;

        ipk = ip;

    }

    void tampilInformasi () {

        System.out.println("Nama      : " + nama);

        System.out.println("Nim      : " + nim);

        System.out.println("Kelas   : " + kelas);

        System.out.println("Ipk      : " + ipk);

        System.out.println("-----");

    }

}

// CLASS MAHASISWA BERPRESTASI

class MahasiswaBerprestasi04 {

    Mahasiswa04 listMhs [] = new Mahasiswa04 [5];

    int idx = 0;

    void tampil () {

        System.out.println("-----");

        for (int i = 0; i < listMhs.length; i++) {

            listMhs[i].tampilInformasi();

        }

    }

}
```

```

    }

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

void tambah(Mahasiswa04 mhs) {
    if (idx < listMhs.length) {
        listMhs[idx] = mhs;
        idx++;
    } else {
        System.out.println("Data penuh");
    }
}

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 index " +
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");
    }
}

```

```

    }
}

class MahasiswaDemo04 {
    public static void main(String[] args) {
        MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();
        Scanner str = new Scanner(System.in);
        int jumMhs = 5;

        for (int i = 0; i < jumMhs; i++) {
            System.out.println("Masukan data mahasiswa ke-" + (i+1));
            System.out.print("Nim   : ");
            String nim = str.nextLine();
            System.out.print("Nama : ");
            String nama = str.nextLine();
            System.out.print("kelas : ");
            String kelas = str.nextLine();
            System.out.print("IPK   : ");
            double ipk = str.nextDouble();
            str.nextLine();

            list.tambah(new Mahasiswa04(nama, nim, kelas, ipk));
        }

        list.tampil();
        // melakukan pencarian data sequential
        System.out.println("-----");
        System.out.println(" Pencarian Data ");
        System.out.println("-----");
        System.out.println("Masukan ipk mahasiswa yang dicari: ");
        System.out.print("IPK: ");
        double cari = str.nextDouble(); str.nextLine();

        System.out.println("Menggunakan sequential searching");
        double posisi = list.sequentialSearching(cari);
        int pss = (int) posisi;
        list.tampilPosisi(cari, pss);
        list.tampilDataSearch(cari, pss);
    }
}

```

```
}  
}
```

```
Nim : hanip  
Nama : 555  
kelas : ti 1a  
IPK : 3,92  
-----  
Nama : fajel  
Nim : 111  
Kelas : ti1a  
Ipk : 3.9  
-----  
Nama : nabel  
Nim : 222  
Kelas : ti 1a  
Ipk : 3.94  
-----  
Nama : king  
Nim : 333  
Kelas : ti 1a  
Ipk : 3.84  
-----  
Nama : preda  
Nim : 444  
Kelas : ti 1a  
Ipk : 3.58  
-----  
Nama : 555  
Nim : hanip  
Kelas : ti 1a  
Ipk : 3.92  
-----  
=====
```

Pencarian Data

```
Masukan ipk mahasiswa yang dicari:  
IPK: 3,58  
Menggunakan sequential searching  
Data mahasiswa dengan IPK : 3.58 ditemukan pada index 3  
Nim : 444  
Nama : preda  
Kelas : ti 1a  
IPK : 3.58  
PS D:\Kuliahh\kuliahhh\Semester2\PrakAlgoritmaStrukturDT> ^C
```

### 6.2.3 Pertanyaan Praktikum 1

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!
  - Pada method tampilDataSearch menampilkan hasil pencarian beserta identitas mahasiswa sedangkan pada metod tampilPosisi tidak menampilkan data mahasiswa
2. Jelaskan fungsi break pada kode program dibawah ini!

- Untuk berhenti melakukan pencarian karena value yang dicari telah ditemukan

## Praktikum 2

```
System.out.println("-----");
    System.out.println(" Pencarian Data ke 2");
    System.out.println("-----");
    System.out.println("Masukan ipk mahasiswa yang dicari: ");
    System.out.print("IPK: ");
    double cari2 = str.nextDouble(); str.nextLine();

    System.out.println("Menggunakan Binary Search");
    int posisi2 = list.sequentialSearching(cari2);
    int pss2 = posisi2;
    list.tampilPosisi(cari, pss2);
    list.tampilDataSearch(cari, pss2);

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;
}
```

1. Tunjukkan pada kode program yang mana proses divide dijalankan!

```

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. Tunjukkan pada kode program yang mana proses conquer dijalankan!

```

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;
}

```

3. Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
- Bisa, hanya saja hasil pencarian akan tidak akurat
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

```

int findBinarySearch (double cari, int left, int right) {
    int mid;
    if (right <= left) {
        mid = (left+right)/2;
        if (cari == listMhs[mid].ipk) {
            return( Mahasiswa04[] listMhs
        } 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.

```

System.out.print(s:"Masukan jumlah seluruh mahasiswa: ");
int jumMhs = str.nextInt(); str.nextLine();
MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04(jumMhs);

```

## Praktikum 2

1. Pada Latihan praktikum pertemuan sebelumnya pada jobsheet 5 yang terdapat 3 class yaitu Dosen, DataDosen , dan DosenDemo, tambahkan method:

- a. PencarianDataSequential : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.

```

int SearchNama (String cari) { // pencarian menggunakan sequentinal Searching
    berdasarkan Nama
    int posisi = -1;
    for (int j = 0; j < dataDosen.length; j++) {
        if (dataDosen[j].nama.equalsIgnoreCase(null)) {
            posisi = j;
            break;
        }
    }
    return posisi;
}

```

- b. PencarianDataBinary : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.

```

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

```



- c. Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan

```
package Pertemuan_6;

import java.util.Scanner;

import Pertemuan_2.dosen04;

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

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

    public void tampil() {
        System.out.println("Kode: " + kode + ", Nama: " + nama + ", Usia: " + usia + ", Jenis
Kelamin: " + (jenisKelamin ? "Laki-laki" : "Perempuan"));
    }
}

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

    public void tambah(Dosen dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data penuh!");
        }
    }

    public void tampil() {
```

```

        for (int i = 0; i < idx; i++) {
            dataDosen[i].tampil();
        }
    }
}

```

```

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

```

```

public void sortingDSC() { // Selection Sort
    for (int i = 0; i < idx - 1; i++) {
        int maxIdx = i;
        for (int j = i + 1; j < idx; j++) {
            if (dataDosen[j].usia > dataDosen[maxIdx].usia) {
                maxIdx = j;
            }
        }
        Dosen temp = dataDosen[maxIdx];
        dataDosen[maxIdx] = dataDosen[i];
        dataDosen[i] = temp;
    }
}

```

```

int SearchNama (String cari) { // pencarian menggunakan sequential Searching berdasarkan
Nama

```

```

    int posisi = -1;
    for (int j = 0; j < dataDosen.length; j++) {
        if (dataDosen[j].nama.equalsIgnoreCase(null)) {
            posisi = j;
            break;
        }
    }
    return posisi;
}

```

```

    }

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

    void tampilPosisi(int x, int pos) {
        if (pos != -1) {
            System.out.println("Data Dosen dengan usia : " + x + " ditemukan pada index " +
pos);
        } else {
            System.out.println("dosen dengan umur " + x + " tidak ditemukan");
        }
    }

    void tampilDataSearch (int x, int pos) {
        if (pos != -1) {
            System.out.println("Kode\t : " + dataDosen[pos].kode);
            System.out.println("Nama\t : " + dataDosen[pos].nama);
            System.out.println("Jenis kelamin\t : " + dataDosen[pos].jenisKelamin);
            System.out.println("umur\t : " + dataDosen[pos].usia);
        } else {
            System.out.println("Data Dosen dengan usia " + x + " tidak ditemukan");
        }
    }
}

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen data = new DataDosen();
    }
}

```

```

int pilihan;

do {

    System.out.println("Menu:");

    System.out.println("1. Tambah Data");

    System.out.println("2. Tampil Data");

    System.out.println("3. Sorting ASC (Usia Muda -> Tua)");

    System.out.println("4. Sorting DSC (Usia Tua -> Muda)");

    System.out.println("5. Mencari data Dosen berdasarkan Nama");

    System.out.println("6. Melakukan pencarian Data dosen berdasarkan umur");

    System.out.println("7. Keluar program");

    System.out.print("Pilih menu: ");

    pilihan = sc.nextInt();

    sc.nextLine(); // Buang newline


    switch (pilihan) {

        case 1:

            System.out.print("Masukkan kode: ");

            String kode = sc.nextLine();

            System.out.print("Masukkan nama: ");

            String nama = sc.nextLine();

            System.out.print("Masukkan jenis kelamin (true = Laki-laki, false =
Perempuan): ");

            boolean jk = sc.nextBoolean();

            System.out.print("Masukkan usia: ");

            int usia = sc.nextInt();

            data.tambah(new Dosen(kode, nama, jk, usia));

            break;

        case 2:

            data.tampil();

            break;

        case 3:

            data.sortingASC();

            System.out.println("Data telah diurutkan secara ASC!");

            break;

        case 4:

            data.sortingDSC();

            System.out.println("Data telah diurutkan secara DSC!");

            break;

        case 5:

            System.out.println("Masukan Nama Dosen yang ingin dicari: ");

            String nm = sc.nextLine();

```

```

        data.SearchNama(nm);
        break;
    case 6:
        System.out.println("Keluar dari program.");
        break;
    case 7 :
        System.out.print("Masukan umur dosen yang ingin dicari: ");
        int find = sc.nextInt(); sc.nextLine();
        data.sortingASC();
        data.findBinarySearch(find, 0, 10-1); // 10 adalah jumlah dosen
    default:
        System.out.println("Pilihan tidak valid!");
    }
} while (pilihan != 6);

sc.close();
}
}

```

<https://github.com/AlfredaDhaifullah04/Semester-2/tree/master/Pertemuan 7>