

**Laporan Hasil Praktikum Algoritma Dan  
Struktur Data  
Jobsheet 7**



**Nama : Arifah Zhafirah Wikananda**

**NIM : 244107020188**

**Kelas : 1E**

**Program Studi Teknologi  
Informasi Jurusan Teknik  
Informatika Politeknik Negeri  
Malang  
2025**

1.

```
package jobsheet7;
public class Mahasiswa04 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa04() {

    }

    Mahasiswa04(String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        ipk = ip;
        kelas = kls;
    }

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

```
package jobsheet7;
public class MahasiswaBerprestasi04 {
    Mahasiswa04 [] listMhs = new Mahasiswa04 [5];
    int idx;

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

    void tampil() {
        for (Mahasiswa04 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) {
                    Mahasiswa04 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;
            }
        }
        Mahasiswa04 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa04 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}

//mulai dari sini jobsheet7
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 +" berada di posisi ke "+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);
        System.out.println("=====");
    } else {
        System.out.println("data mahasiswa dengan IPK "+ x +" tidak ditemukan");
    }
}
}

```

```

package jobsheet7;
import java.util.Scanner;
public class MahasiswaDemo04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int jumMhs = 5;
        MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();

        for (int i = 0; i < jumMhs; i++) {
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");
            System.out.print("NIM: ");
            String nim = sc.nextLine();
            System.out.print("Nama: ");
            String nama = sc.nextLine();
            System.out.print("Kelas: ");
            String kelas = sc.nextLine();
            System.out.print("IPK: ");
            String ip = sc.nextLine();
            double ipk = Double.parseDouble(ip);
            System.out.println("-----");
            list.tambah (new Mahasiswa04(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);
        sc.close();
    }
}

```

Hasil dari langkah percobaan diatas:

```
C:\Praktikum ASD> c: && cd "c:\Praktikum ASD" &&
ges -cp "C:\Users\WINDOWS 11\AppData\Roaming\Microsoft\Windows\CurrentVersion\Temp\c51dcdc\bin" jobsheet7.MahasiswaDemo04 "
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: 3.5
-----
```

```
-----
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 berada di posisi ke 3
nim      : 444
nama     : lia
kelas    : 2
ipk      : 3.5
=====
```

### 7.2.3. Pertanyaan

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!

Jawab:

1. tampilDataSearch untuk memberikan sebuah informasi lengkap tentang mahasiswa berdasarkan IPK.
2. tampilPosisi untuk memberikan informasi tentang posisi mahasiswa berdasarkan IPK.

2. Jelaskan fungsi break pada kode program dibawah ini!

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

Jawab: berfungsi untuk menghentikan perulangan lebih awal ketika kondisi if terpenuhi.

2.

#### **MahasiswaBerprestasi04.java**

```
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;
}
```

#### **MahasiswaDemo04.java**

```
//melakukan pencarian data binary
System.out.println("-----");
System.out.println("Pencarian data");
System.out.println("-----");
System.out.println("masukkan ipk mahasiswa yang dicari: ");
System.out.print("IPK: ");
double cari2 = sc.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);
```

Hasil dari langkah percobaan diatas:

```
c:\Praktikum ASD>
c:\Praktikum ASD> c: && cd "c:\Praktikum ASD"
ges -cp "C:\Users\WINDOWS 11\AppData\Roaming\Microsoft\Windows\CurrentVersion\Templates\c51dcdc\bin" jobsheet7.MahasiswaDemo04 "
Masukkan data mahasiswa ke-1:
NIM: 111
Nama: adi
Kelas: 2
IPK: 3.2
-----
Masukkan data mahasiswa ke-2:
NIM: 222
Nama: lia
Kelas: 2
IPK: 3.1
-----
Masukkan data mahasiswa ke-3:
NIM: 333
Nama: ila
Kelas: 2
IPK: 3.3
-----
Masukkan data mahasiswa ke-4:
NIM: 444
Nama: susi
Kelas: 2
IPK: 3.4
-----
Masukkan data mahasiswa ke-5:
NIM: 555
Nama: afta
Kelas: 2
IPK: 3.5
-----
Nama : adi
NIM : 111
-----
Masukkan IPK mahasiswa yang dicari:
IPK : 3.4
-----
menggunakan Sequential Search
-----
data mahasiswa dengan IPK 3.4 berada di posisi ke 3
nim      : 444
nama     : susi
kelas    : 2
ipk      : 3.4
=====
-----
menggunakan binary search
-----
data mahasiswa dengan IPK 3.4 berada di posisi ke 3
nim      : 444
nama     : susi
kelas    : 2
ipk      : 3.4
=====
```

```
-----
Nama : adi
NIM : 111
Kelas: 2
IPK : 3.2
-----
Nama : lia
NIM : 222
Kelas: 2
IPK : 3.1
-----
Nama : ila
NIM : 333
Kelas: 2
IPK : 3.3
-----
Nama : susi
NIM : 444
Kelas: 2
IPK : 3.4
-----
Nama : afta
NIM : 555
Kelas: 2
IPK : 3.5
-----
Pencarian data
-----
Masukkan IPK mahasiswa yang dicari:
IPK : 3.4
-----
menggunakan Sequential Search
-----
data mahasiswa dengan IPK 3.4 berada di posisi ke 3
nim      : 444
```



### 7.3.3. 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!

Jawab: Jika data IPK yang dimasukkan tidak urut maka program tetap akan berjalan tetapi tidak berjalan semestinya, karena jika kita menggunakan binary search data harus urut tetapi jika kita menggunakan sequential search masih berfungsi tetapi tidak akan memberikan hasil yang akurat.

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

Jawab: Tidak sesuai karena algoritma nya menggunakan urutan ascending.

```
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);  
}
```

Ubah ke arah > untuk menggunakan urutan descending

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

```
public static void main(String[] args) {  
    MahasiswaBerprestasi04 list = new MahasiswaBerprestasi04();  
    Scanner sc = new Scanner(System.in);  
    System.out.println(x:"Masukkan jumlah mahasiswa: ");  
    int jumMhs = sc.nextInt();  
    sc.nextLine();  
}
```

```
Masukkan jumlah mahasiswa:  
5  
Masukkan data mahasiswa ke-1:  
NIM: 111
```

## LATIHAN

```
package jobsheet7;
public class Dosen04 {
    String kode;
    String nama;
    Boolean jenisKelamin;
    int usia;

    Dosen04(){
    }

    Dosen04(String kd, String name, Boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

    void tampil(){
        System.out.println("Kode : " + kode);
        System.out.println("Nama : " + nama);
        System.out.println("Jenis kelamin : " + jenisKelamin);
        System.out.println("Umur : " + usia);
    }
}
```

```
package jobsheet7;
public class DataDosen04 {
    Dosen04[] dataDosen = new Dosen04[10];
    int idx;

    void tambah(Dosen04 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data dosen sudah penuh!");
        }
    }

    void tampil() {
        if (idx == 0) {
            System.out.println("Tidak ada data dosen.");
            return;
        }
        for (int i = 0; i < idx; i++) {
            System.out.println("Posisi ke-" + i);
            dataDosen[i].tampil();
            System.out.println();
        }
    }
}
```

```

void sortingASC() {
    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) {
                Dosen04 temp = dataDosen[j];
                dataDosen[j] = dataDosen[j + 1];
                dataDosen[j + 1] = temp;
            }
        }
    }
}

void sortingDSC() {
    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) {
                Dosen04 temp = dataDosen[j];
                dataDosen[j] = dataDosen[j + 1];
                dataDosen[j + 1] = temp;
            }
        }
    }
}

void PencarianDataSequential(String nama) {
    boolean found = false;
    int count = 0;
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(nama)) {
            System.out.println("Data dosen dengan nama dosen "+ nama +" berada di posisi ke-" + i);
            dataDosen[i].tampil();
            found = true;
            count++;
        }
    }
    if (!found) {
        System.out.println("Data dosen dengan nama " + nama + " tidak ditemukan.");
    } else if (count > 1) {
        System.out.println("Peringatan: Terdapat lebih dari satu dosen dengan nama " + nama + ".");
    }
}

void PencarianDataBinary(int usia) {
    sortingASC();
    int left = 0, right = idx - 1;
    boolean found = false;
    int count = 0;

    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (dataDosen[mid].usia == usia) {
            int i = mid;
            while (i >= left && dataDosen[i].usia == usia) {
                System.out.println("Data dosen dengan usia dosen "+ usia +" berada di posisi ke-" + i);
                dataDosen[i].tampil();
                System.out.println("Usia: " + dataDosen[i].usia);
                found = true;
                count++;
                i--;
            }
            i = mid + 1;
        }
    }
}

```

```

        while (i <= right && dataDosen[i].usia == usia) {
            System.out.println("Data dosen dengan usia dosen " + usia + " berada di
posisi ke-" + i);
            dataDosen[i].tampil();
            System.out.println("Usia: " + dataDosen[i].usia);
            found = true;
            count++;
            i++;
        }
        break;
    } else if (dataDosen[mid].usia < usia) {
        left = mid + 1;
    } else {
        right = mid - 1;
    }
}

if (!found) {
    System.out.println("Data dosen dengan usia " + usia + " tidak ditemukan.");
} else if (count > 1) {
    System.out.println("Peringatan: Terdapat lebih dari satu dosen dengan usia " +
usia + ".");
}
}
}

```

```

package jobsheet7;
import java.util.Scanner;
public class DosenDemo04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen04 dataDsn = new DataDosen04();
        int pilihan;

        do {
            System.out.println("\nMenu");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Sorting ASC (Usia termuda ke tertua)");
            System.out.println("4. Sorting DSC (Usia tertua ke termuda)");
            System.out.println("5. Cari Dosen berdasarkan Nama (Sequential Search)");
            System.out.println("6. Cari Dosen berdasarkan Usia (Binary Search)");
            System.out.println("7. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Masukkan jumlah dosen yang ingin ditambahkan: ");
                    int jumlah = sc.nextInt();
                    sc.nextLine();
                    for (int i = 0; i < jumlah; i++) {
                        System.out.println("Masukkan data dosen ke-" + (i + 1) + ":");
                        System.out.print("Masukkan kode dosen: ");
                        String kode = sc.nextLine();
                        System.out.print("Masukkan nama dosen: ");
                        String nama = sc.nextLine();
                        System.out.print("Masukkan jenis kelamin (L/P): ");
                        char jk = sc.next().charAt(0);
                        Boolean jenisKelamin = (jk == 'L' || jk == 'l');

```

```

        System.out.print("Masukkan usia dosen: ");
        int usia = sc.nextInt();
        sc.nextLine();
        System.out.println("-----");

        Dosen04 dsn = new Dosen04(kode, nama, jenisKelamin, usia);
        dataDsn.tambah(dsn);
    }
    break;
case 2:
    dataDsn.tampil();
    break;
case 3:
    dataDsn.sortingASC();
    System.out.println("Data dosen telah diurutkan secara ascending.");
    dataDsn.tampil();
    break;
case 4:
    dataDsn.sortingDSC();
    System.out.println("Data dosen telah diurutkan secara descending.");
    dataDsn.tampil();
    break;
case 5:
    System.out.print("Masukkan nama dosen yang ingin dicari: ");
    String cariNama = sc.nextLine();
    dataDsn.PencarianDataSequential(cariNama);
    break;
case 6:
    System.out.print("Masukkan usia dosen yang ingin dicari: ");
    int cariUsia = sc.nextInt();
    sc.nextLine();
    dataDsn.PencarianDataBinary(cariUsia);
    break;
case 7:
    System.out.println("Keluar dari program.");
    break;
default:
    System.out.println("Pilihan tidak valid. Silakan coba lagi.");
    }
} while (pilihan != 7);
}
}

```

## Hasil dari langkah diatas:

```
C:\Praktikum ASD> cmd /C ""C:\Program Files\Java\jdk-2
cp "C:\Users\WINDOWS 11\AppData\Roaming\Code\User\work
\jdt_ws\Praktikum ASD_c51dcdc\bin" jobsheet7.DosenDemo
```

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia termuda ke tertua)
4. Sorting DSC (Usia tertua ke termuda)
5. Cari Dosen berdasarkan Nama (Sequential Search)
6. Cari Dosen berdasarkan Usia (Binary Search)
7. Keluar

Pilih menu: 1

Masukkan jumlah dosen yang ingin ditambahkan: 3

Masukkan data dosen ke-1:

Masukkan kode dosen: 234

Masukkan nama dosen: kayla

Masukkan jenis kelamin (L/P): 1

Masukkan usia dosen: 23

-----

Masukkan data dosen ke-2:

Masukkan kode dosen: 345

Masukkan nama dosen: layla

Masukkan jenis kelamin (L/P): p

Masukkan usia dosen: 43

-----

Masukkan data dosen ke-3:

Masukkan kode dosen: 456

Masukkan nama dosen: jay

Masukkan jenis kelamin (L/P): 1

Masukkan usia dosen: 35

-----

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia termuda ke tertua)
4. Sorting DSC (Usia tertua ke termuda)
5. Cari Dosen berdasarkan Nama (Sequential Search)
6. Cari Dosen berdasarkan Usia (Binary Search)
7. Keluar

Pilih menu: 5

Masukkan nama dosen yang ingin dicari: kayla

Data dosen dengan nama dosen kayla berada di posisi ke-2

Kode : 234

Nama : kayla

Jenis kelamin : true

Umur : 23

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia termuda ke tertua)
4. Sorting DSC (Usia tertua ke termuda)
5. Cari Dosen berdasarkan Nama (Sequential Search)
6. Cari Dosen berdasarkan Usia (Binary Search)
7. Keluar

Pilih menu: 6

Masukkan usia dosen yang ingin dicari: 35

Data dosen dengan usia dosen 35 berada di posisi ke-1

Kode : 456

Nama : jay

Jenis kelamin : true

Umur : 35

Usia: 35

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia termuda ke tertua)
4. Sorting DSC (Usia tertua ke termuda)
5. Cari Dosen berdasarkan Nama (Sequential Search)
6. Cari Dosen berdasarkan Usia (Binary Search)
7. Keluar

Pilih menu: 2

Posisi ke-0

Kode : 234

Nama : kayla

Jenis kelamin : true

Umur : 23

Posisi ke-1

Kode : 345

Nama : layla

Jenis kelamin : false

Umur : 43

Posisi ke-2

Kode : 456

Nama : jay

Jenis kelamin : true

Umur : 35

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia termuda ke tertua)
4. Sorting DSC (Usia tertua ke termuda)
5. Cari Dosen berdasarkan Nama (Sequential Search)
6. Cari Dosen berdasarkan Usia (Binary Search)
7. Keluar

Pilih menu: 4

Data dosen telah diurutkan secara descending.

Posisi ke-0

Kode : 345

Nama : layla

Jenis kelamin : false

Umur : 43

Posisi ke-1

Kode : 456

Nama : jay

Jenis kelamin : true

Umur : 35

Posisi ke-2

Kode : 234

Nama : kayla

Jenis kelamin : true

Umur : 23

Menu

1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia termuda ke tertua)
4. Sorting DSC (Usia tertua ke termuda)
5. Cari Dosen berdasarkan Nama (Sequential Search)
6. Cari Dosen berdasarkan Usia (Binary Search)
7. Keluar

Pilih menu: 3

Data dosen telah diurutkan secara ascending.

Posisi ke-0

Kode : 234

Nama : kayla

Jenis kelamin : true

Umur : 23

Posisi ke-1

Kode : 456

Nama : jay

Jenis kelamin : true

Umur : 35

Posisi ke-2

Kode : 345

Nama : layla

Jenis kelamin : false

Umur : 43