

Niky Tania Sari (18/SIB-1B)

## PERTEMUAN 7

### JOBSHEET 6 SEARCHING

#### Praktikum 1 Searching / Pencarian Menggunakan Algoritma Sequential Search

```
public class Mahasiswa {  
    String nim;  
    String nama;  
    String kelas;  
    double ipk;  
  
    //konstruktor default  
    Mahasiswa() {  
    }  
  
    //konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama ada yang tidak)  
    Mahasiswa(String nm, String name, double ip, String kls){  
        nim = nm;  
        nama = name;  
        ipk = ip;  
        kelas = kls;  
    }  
    void tampilInformasi(){  
        System.out.println("Nama: " + nama);  
        System.out.println("NIM: " + nim);  
        System.out.println("IPK: " + ipk);  
        System.out.println("Kelas: " + kelas);  
        System.out.println(x:"_____");  
    }  
}
```

```

MahasiswaBerprestasi18.java

import javax.xml.transform.Source;

public class MahasiswaBerprestasi18 {
    Mahasiswa [] listMhs = new Mahasiswa [5];
    int idx;

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

    void tampil () {
        for (Mahasiswa m: listMhs) {
            m.tampilInformasi();
        }
    }

    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) {
                    Mahasiswa 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) { //ascending
                    idxMin = j;
                }
            }
            Mahasiswa tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }

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

    // Sequential Search Method
    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");
        }
    }
}

```

```

import java.util.Scanner;

public class MahasiswaDemo18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        MahasiswaBerprestasi18 list = new MahasiswaBerprestasi18();
        Scanner sc = new Scanner(System.in);
        int jumlahMahasiswa=5;

        for (int i = 0; i < jumlahMahasiswa; i++) {
            System.out.println("\nMasukkan 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   : ");
            String ipk = sc.nextLine();
            Double ipk = Double.parseDouble(ipk);
            System.out.println(x:"-----");

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

        list.tampil();
        //melakukan pencarian data sequential
        System.out.println(x:"-----");
        System.out.println(x:"Pencarian Data");
        System.out.println(x:"-----");
        System.out.print(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);
    }
}

```

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

```
Masukkan data mahasiswa ke-5
```

```
NIM : 555
```

```
Nama : fia
```

```
Kelas : 2
```

```
IPK : 3.3
```

```
-----
```

```
Nama: adi
```

```
NIM: 111
```

```
IPK: 3.6
```

```
Kelas: 2
```

```
-----
```

```
Nama: tio
```

```
NIM: 222
```

```
IPK: 3.8
```

```
Kelas: 2
```

```
-----
```

```
Nama: ila
```

```
NIM: 333
```

```
IPK: 3.0
```

```
Kelas: 2
```

```
-----
```

```
Nama: lia
```

```
NIM: 444
```

```
IPK: 3.5
```

```
Kelas: 2
```

```
-----
```

```
Nama: fia
```

```
NIM: 555
```

```
IPK: 3.3
```

```
NIM: 555
```

```
IPK: 3.3
```

```
Kelas: 2
```

```
-----
```

```
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!

**Jawaban:**

- **tampilDataSearch(double x, int pos)**

Metode ini digunakan untuk menampilkan detail informasi mahasiswa berdasarkan posisi (indeks) yang ditemukan dari pencarian. Jika data tidak ditemukan, akan menampilkan pesan bahwa data mahasiswa dengan IPK tersebut tidak ditemukan.

**Contoh output:**

nim : 22031234

nama : Adi

kelas : TI-2B

ipk : 3.75

- **tampilPosisi(double x, int pos)**

Metode ini digunakan untuk menampilkan posisi (indeks) dari mahasiswa yang memiliki IPK yang dicari. Jika data ditemukan, akan dicetak posisi indeksinya. Jika tidak ditemukan, akan menampilkan pesan bahwa data tidak ditemukan.

**Contoh output:**

data mahasiswa dengan IPK: 3.75 ditemukan pada indeks 2

2. Jelaskan fungsi break pada kode program dibawah ini!

**Jawaban:**

- Fungsi break adalah untuk menghentikan perulangan segera setelah menemukan mahasiswa dengan IPK yang dicari.
- Jika ditemukan, variabel posisi akan menyimpan indeks mahasiswa tersebut, dan break akan keluar dari loop for, sehingga program tidak perlu mengecek elemen lainnya.

## Praktikum 2

```
public class Mahasiswa {
    String nim;
    String nama;
    String kelas;
    double ipk;

    //konstruktor default
    Mahasiswa() {
    }

    //konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama ada yang tidak)
    Mahasiswa(String nm, String name, double ip, String kls){
        nim = nm;
        nama = name;
        ipk = ip;
        kelas = kls;
    }

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

```

MahasiswaBerprestasi18.java

public class MahasiswaBerprestasi18 {
    Mahasiswa [] listMhs = new Mahasiswa [5];
    int idx;

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

    void tampil () {
        for (Mahasiswa m: listMhs) {
            m.tampilInformasi();
        }
    }

    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) {
                    Mahasiswa 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) { //ascending
                    idxMin = j;
                }
            }
            Mahasiswa tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }

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

    // Sequential Search Method
    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");
        }
    }

    // Binary Search Method
    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;
    }
}

```

```

import java.util.Scanner;

public class MahasiswaDemo18 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        MahasiswaBerprestasi18 list = new MahasiswaBerprestasi18();
        Scanner sc = new Scanner(System.in);
        int jumlahMahasiswa=5;

        for (int i = 0; i < jumlahMahasiswa; i++) {
            System.out.println("\nMasukkan 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   : ");
            String ip = sc.nextLine();
            Double ipk = Double.parseDouble(ip);
            System.out.println(x:"-----");

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

            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:"-----");
            System.out.println(x:"menggunakan binary search");
            System.out.println(x:"-----");
            double posisi2 = list.findBinarySearch(cari, left:0, jumlahMahasiswa-1);
            int pss2 = (int)posisi2;
            list.tampilPosisi(cari, pss2);
            list.tampilDataSearch(cari,pss2);
        }
    }
}

```

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



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

IPK: 3.1

Kelas: 2

-----

Nama: ila

NIM: 222

IPK: 3.2

Kelas: 2

-----

Nama: lia

NIM: 333

IPK: 3.3

Kelas: 2

-----

Nama: susi

Nama: susi

NIM: 444

IPK: 3.5

Kelas: 2

-----

Nama: anita

NIM: 555

IPK: 3.7

Kelas: 2

-----

-----

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

press any key to continue

## Pertanyaan

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

**Jawaban:** Proses divide dalam binary search terjadi saat array dibagi dua dalam setiap rekursi. Terjadi pada method `findBinarySearch` saat menghitung  $mid = (left + right) / 2$ .

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

2. Tunjukkan pada kode program yang mana proses conquer dijalankan!

**Jawaban:** Proses conquer dalam binary search terjadi saat program memeriksa apakah nilai yang dicari (*cari*) ditemukan pada indeks tengah (*mid*). Jika iya, indeks tersebut dikembalikan. Jika tidak, pencarian dilanjutkan ke kiri (*left, mid-1*) atau kanan (*mid+1, right*) melalui rekursi. Ini terjadi dalam method `findBinarySearch`

```
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);
}
```

3. Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!

**Jawaban:** Program tidak akan berjalan dengan benar untuk binary search karena binary search membutuhkan data yang sudah terurut. Jika data tidak urut, algoritma tidak dapat menentukan apakah harus mencari di bagian kiri atau kanan array

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

**Jawaban:** Jika IPK diurutkan dari terbesar ke terkecil (*descending*) dan mencari IPK 3.2, hasil binary search tidak akan sesuai karena kode binary search saat ini mengasumsikan data *ascending* (kecil ke besar), sementara metode `bubbleSort()` dan `insertionSort()` mengurutkan secara *descending*. Akibatnya, perbandingan `listMhs[mid].ipk > cari` mengarahkan pencarian ke arah yang salah.

Modifikasi

```

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);
        }
    }
}

```

Tanda > menjadi <

Lalu setelah diubah, pencarian binary search akan sesuai dengan data yang diurutkan secara descending.

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

**Jawaban:**

```

MahasiswaBerprestasi list = new MahasiswaBerprestasi();
Scanner sc = new Scanner(System.in);

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

list.listMhs = new Mahasiswa[jumlahMahasiswa];

```

## Latihan

```
public class Dosen18 {  
    String kode;  
    String nama;  
    Boolean jenisKelamin;  
    int usia;  
  
    Dosen18(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 ? "L" : "P"));  
        System.out.println("Usia: " + usia);  
        System.out.println(x: "_____");  
    }  
}
```

```

DataDosen18.java

public class DataDosen18 {
    Dosen18[] dataDosen = new Dosen18[10];
    int idx = 0;

    public void tambah(Dosen18 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh!");
        }
    }

    public void tampil() {
        if (idx == 0) {
            System.out.println("Tidak ada data dosen!");
            return;
        }
        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 - i - 1; j++) {
                if (dataDosen[j].usia > dataDosen[j + 1].usia) {
                    Dosen18 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
    }

    public void sortingDESC() { // Selection Sort
        for (int i = 0; i < idx - 1; i++) {
            int idxMax = i;
            for (int j = i + 1; j < idx; j++) {
                if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                    idxMax = j;
                }
            }
            Dosen18 temp = dataDosen[idxMax];
            dataDosen[idxMax] = dataDosen[i];
            dataDosen[i] = temp;
        }
    }

    public void insertionSort() {
        for (int i = 1; i < idx; i++) {
            Dosen18 key = dataDosen[i];
            int j = i - 1;
            while (j >= 0 && dataDosen[j].usia < key.usia) {
                dataDosen[j + 1] = dataDosen[j];
                j--;
            }
            dataDosen[j + 1] = key;
        }
    }

    public void PencarianDataSequential18(String namaCari){
        boolean ditemukan = false;
        int count = 0;

        System.out.println("\nHasil Pencarian Sequential berdasarkan Nama: ");
        for (int i = 0; i < idx; i++){
            if (dataDosen[i].nama.equalsIgnoreCase(namaCari)){
                dataDosen[i].tampil();
                ditemukan = true;
                count++;
            }
        }
        if (!ditemukan){
            System.out.println("Dosen dengan nama " + namaCari+ " tidak ditemukan");
        } else if (count > 1) {
            System.out.println("Peringatan! Ditemukan lebih dari satu dosen dengan nama " + namaCari);
        }
    }

    public int binarySearch(int usiaCari, int left, int right){
        if (right >= left) {
            int mid = (left + right) / 2;
            if (dataDosen[mid].usia == usiaCari) {
                return mid;
            }
            if (dataDosen[mid].usia > usiaCari){
                return binarySearch(usiaCari, left, mid-1);
            }
            return binarySearch(usiaCari, mid+1, right);
        }
        return -1;
    }

    public void PencarianDataBinary(int usiaCari) {
        sortingASC(); // Binary search butuh data terurut ascending
        int hasil = binarySearch(usiaCari, 0, idx - 1);

        if (hasil != -1){
            System.out.println("\nHasil Pencarian Binary berdasarkan Usia:");
            dataDosen[hasil].tampil();

            // Cek jika ada usia yang sama
            int left = hasil - 1, right = hasil + 1;
            boolean adaLebihDariSatu = false;

            while (left >= 0 && dataDosen[left].usia == usiaCari){
                dataDosen[left].tampil();
                adaLebihDariSatu = true;
                left--;
            }
            while (right < idx && dataDosen[right].usia == usiaCari) {
                dataDosen[right].tampil();
                adaLebihDariSatu = true;
                right++;
            }

            if (adaLebihDariSatu) {
                System.out.println("Peringatan! Ditemukan lebih dari satu dosen dengan usia " + usiaCari);
            }
        } else {
            System.out.println("Dosen dengan usia " + usiaCari + " tidak ditemukan.");
        }
    }
}

```

```

DosenMain18.java

import java.util.Scanner;

public class DosenMain18 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen18 data = new DataDosen18();

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Tambah data dosen");
            System.out.println("2. Tampil data dosen");
            System.out.println("3. Sorting ASC (Bubble Sort)");
            System.out.println("4. Sorting DSC (Selection Sort)");
            System.out.println("5. Sorting DSC (Insertion Sort)");
            System.out.println("6. Pencarian Sequential (Nama)");
            System.out.println("7. Pencarian Binary (Usia)");
            System.out.println("8. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    while (true) { //perulangan untuk menambah lebih dari satu dosen
                        System.out.println("-----");
                        System.out.print("Masukkan kode dosen: ");
                        String kode = sc.nextLine();
                        System.out.print("Masukkan nama dosen: ");
                        String nama = sc.nextLine();
                        System.out.print("Jenis Kelamin (L/P): ");
                        boolean jk = sc.nextLine().trim().equalsIgnoreCase("L");
                        System.out.print("Masukkan usia dosen: ");
                        int usia = sc.nextInt();
                        sc.nextLine();

                        Dosen18 dsn = new Dosen18(kode, nama, jk, usia);
                        data.tambah(dsn);

                        System.out.print("Tambah dosen lagi? (Y/N): ");
                        String lanjut = sc.nextLine().trim();
                        if (lanjut.equalsIgnoreCase("N")) {
                            break; // Keluar dari perulangan tambah data
                        }
                    }
                    break;
                case 2:
                    data.tampil();
                    break;
                case 3:
                    data.sortingASC();
                    System.out.println("Data dosen telah diurutkan secara ASCENDING (Usia termuda ke tertua)");
                    data.tampil();
                    break;
                case 4:
                    data.sortingDSC();
                    System.out.println("Data dosen telah diurutkan secara DESCENDING (Selection Sort) Usia tertua ke termuda");
                    data.tampil();
                    break;
                case 5:
                    data.insertionSort();
                    System.out.println("Data dosen telah diurutkan secara DESCENDING (Insertion Sort)");
                    data.tampil();
                    break;
                case 6:
                    System.out.print("Masukkan nama dosen yang dicari: ");
                    String namaCari = sc.nextLine();
                    data.PencarianDataSequential18(namaCari);
                    break;
                case 7:
                    System.out.print("Masukkan usia dosen yang dicari: ");
                    int usiaCari = sc.nextInt();
                    sc.nextLine(); // Buang newline
                    data.PencarianDataBinary(usiaCari);
                    break;
                case 8:
                    System.out.println("Keluar dari program");
                    return; // Hentikan program
                default:
                    System.out.println("Pilihan tidak valid! Coba lagi.");
            }
        }
    }
}

```

```
Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Pencarian Sequential (Nama)
7. Pencarian Binary (Usia)
8. Keluar
Pilih menu: 1
-----
Masukkan kode dosen: 111
Masukkan nama dosen: kia
Jenis Kelamin (L/P): P
Masukkan usia dosen: 30
Tambah dosen lagi? (Y/N): Y
-----
Masukkan kode dosen: 222
Masukkan nama dosen: vina
Jenis Kelamin (L/P): P
Masukkan usia dosen: 30
Tambah dosen lagi? (Y/N): Y
-----
Masukkan kode dosen: 333
Masukkan nama dosen: bion
```

```
Masukkan nama dosen: bion
Jenis Kelamin (L/P): L
Masukkan usia dosen: 45
Tambah dosen lagi? (Y/N): Y
-----
Masukkan kode dosen: 444
Masukkan nama dosen: bion
Jenis Kelamin (L/P): L
Masukkan usia dosen: 40
Tambah dosen lagi? (Y/N): Y
-----
Masukkan kode dosen: 555
Masukkan nama dosen: vivi
Jenis Kelamin (L/P): P
Masukkan usia dosen: 35
Tambah dosen lagi? (Y/N): N

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Pencarian Sequential (Nama)
```

```
6. Pencarian Sequential (Nama)
7. Pencarian Binary (Usia)
8. Keluar
Pilih menu: 6
Masukkan nama dosen yang dicari: bion

Hasil Pencarian Sequential berdasarkan Nama:
Kode: 333
Nama: bion
Jenis Kelamin: L
Usia: 45
-----
Kode: 444
Nama: bion
Jenis Kelamin: L
Usia: 40
-----
Peringatan! Ditemukan lebih dari satu dosen dengan nama bion

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
```

```
4. Sorting DSC (Selection Sort)
5. Sorting DSC (Insertion Sort)
6. Pencarian Sequential (Nama)
7. Pencarian Binary (Usia)
8. Keluar
Pilih menu: 7
Masukkan usia dosen yang dicari: 30

Hasil Pencarian Binary berdasarkan Usia:
Kode: 111
Nama: kia
Jenis Kelamin: P
Usia: 30
-----
Kode: 222
Nama: vina
Jenis Kelamin: P
Usia: 30
-----
Peringatan! Ditemukan lebih dari satu dosen dengan usia 30

Menu:
1. Tambah data dosen
2. Tampil data dosen
3. Sorting ASC (Bubble Sort)
4. Sorting DSC (Selection Sort)
```