

POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA



Nama	: Muhammad Nuril Huda
Kelas	: TI-1A
No	: 19
Mata Kuliah	: Algoritma dan Struktur Data

6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

6.2.1 Kode Program

```
package Pertemuan6;

import java.util.Scanner;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jumlah = 5;

        MahasiswaBerprestasi19 list = new MahasiswaBerprestasi19(jumlah);

        for (int i = 0; i < jumlah; i++) {

            System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1));

            System.out.print("Nama: ");

            String nama = sc.nextLine();

            System.out.print("NIM: ");

            String nim = sc.nextLine();

            System.out.print("Kelas: ");

            String kelas = sc.nextLine();

            System.out.print("IPK: ");

            double ipk = sc.nextDouble();

            sc.nextLine();

            list.tambah(new Mahasiswa19(nama, nim, kelas, ipk));

        }

        list.tampil();

        //melakukan pencarian data sequential

        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);

    }

}
```

```

package Pertemuan6;

public class MahasiswaBerprestasi19 {

    Mahasiswa19[] listMhs;

    int idx;

    MahasiswaBerprestasi19(int jumlah) {

        listMhs = new Mahasiswa19[jumlah];

        idx = 0;

    }

    void tampil () {

        for (Mahasiswa19 m : listMhs) {

            m.tampilInformasi();

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

        }

    }

    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");

        }

    }

}

```

6.2.2 Hasil Percobaan

```
Masukkan data mahasiswa ke-1
Nama: 111
NIM: Adi
Kelas: 2
IPK: 3.6

Masukkan data mahasiswa ke-2
Nama: 222
NIM: tio
Kelas: 2
IPK: 3.8

Masukkan data mahasiswa ke-3
Nama: 333
NIM: ila
Kelas: 2
IPK: 3

Masukkan data mahasiswa ke-4
Nama: 444
NIM: lia
Kelas: 2
IPK: 3.5

Masukkan data mahasiswa ke-5
Nama: 555
NIM: 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 ditemukan pada indeks 3
nim      : 444
nama     : lia
kelas    : 2
ipk      : 3.5
```

6.2.1 Pertanyaan

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!
 - tampilPosisi
 - Metode ini hanya menampilkan posisi (indeks) mahasiswa dalam array listMhs berdasarkan nilai IPK (x) yang dicari
 - Jika mahasiswa dengan IPK tersebut ditemukan, maka akan dicetak posisi indeks nya
 - Jika tidak ditemukan, akan mencetak pesan bahwa data tidak ditemukan
 - tampilDataSearch
 - Metode ini menampilkan seluruh informasi mahasiswa yang memiliki IPK (x) yang dicari
 - Jika ditemukan, akan menampilkan NIM, Nama, Kelas, dan IPK mahasiswa tersebut
 - Jika tidak ditemukan, akan mencetak pesan bahwa data mahasiswa dengan IPK yang dicari tidak ada
2. Jelaskan fungsi break pada kode program dibawah ini!

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

- Break berfungsi untuk menghentikan perulangan setelah menemukan nilai yang cocok, sehingga program tidak perlu mengecek semua elemen dalam array

6.3. Searching / Pencarian Menggunakan Binary Search

6.3.1 Kode Program

```
package Pertemuan6;

public class MahasiswaBerprestasi19 {

    Mahasiswa19[] listMhs;

    int idx;

    MahasiswaBerprestasi19(int jumlah) {

        listMhs = new Mahasiswa19[jumlah];

        idx = 0;

    }

    void tampil () {

        for (Mahasiswa19 m : listMhs) {

            m.tampilInformasi();

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

        }

    }

    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;

    }

}
```

```

package Pertemuan6;

import java.util.Scanner;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jumlah = 5;

        MahasiswaBerprestasi19 list = new MahasiswaBerprestasi19(jumlah);

        for (int i = 0; i < jumlah; i++) {

            System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1));

            System.out.print("Nama: ");

            String nama = sc.nextLine();

            System.out.print("NIM: ");

            String nim = sc.nextLine();

            System.out.print("Kelas: ");

            String kelas = sc.nextLine();

            System.out.print("IPK: ");

            double ipk = sc.nextDouble();

            sc.nextLine();

            list.tambah(new Mahasiswa19(nama, nim, kelas, ipk));

        }

        list.tampil();

        //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 cari = sc.nextDouble();

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

        System.out.println("menggunakan binary search");

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

        double posisi2 = list.findBinarySearch(cari, 0, jumlah-1);

        int pss2 = (int)posisi2;

        list.tampilPosisi(cari, pss2);

        list.tampilDataSearch(cari, pss2);

    }

}

```

6.3.2 Hasil Percobaan

```
Masukkan data mahasiswa ke-1
Nama: huda
NIM: 111
Kelas: 2
IPK: 3.1

Masukkan data mahasiswa ke-2
Nama: ahud
NIM: 222
Kelas: 2
IPK: 3.2

Masukkan data mahasiswa ke-3
Nama: nuril
NIM: 333
Kelas: 2
IPK: 3.3

Masukkan data mahasiswa ke-4
Nama: lirun
NIM: 444
Kelas: 2
IPK: 3.5

Masukkan data mahasiswa ke-5
Nama: ril
NIM: 555
Kelas: 2
IPK: 3.7
```

```
Nama: huda
NIM: 111
Kelas: 2
IPK: 3.1
```

```
-----
Nama: ahud
NIM: 222
Kelas: 2
IPK: 3.2
```

```
-----
Nama: nuril
NIM: 333
Kelas: 2
IPK: 3.3
```

```
-----
Nama: lirun
NIM: 444
Kelas: 2
IPK: 3.5
```

```
-----
Nama: ril
NIM: 555
Kelas: 2
IPK: 3.7
-----
```

```
-----
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     : ril
kelas    : 2
ipk      : 3.7
```


6.3.3 Pertanyaan

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

```
int mid;

if (right>=left){

    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 tidakurut. Apakah program masih dapat berjalan? Mengapa demikian!
 - Jika data IPK yang dimasukkan tidakurut, maka program tidak akan bekerja dengan benar. Karena Binary Search hanya bisa bekerja dengan benar jika data sudah dalam keadaan terurut
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
 - Tidak sesuai
 - Mengubah tanda > menjadi <

```
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);

    }

}
```

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

```
package Pertemuan6;

import java.util.Scanner;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan Jumlah Mahasiswa: ");

        int jumlah = sc.nextInt();

        sc.nextLine();

        MahasiswaBerprestasi19 list = new MahasiswaBerprestasi19(jumlah);

        for (int i = 0; i < jumlah; i++) {

            System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1));

            System.out.print("Nama: ");

            String nama = sc.nextLine();

            System.out.print("NIM: ");

            String nim = sc.nextLine();

            System.out.print("Kelas: ");

            String kelas = sc.nextLine();

            System.out.print("IPK: ");

            double ipk = sc.nextDouble();

            sc.nextLine();

            list.tambah(new Mahasiswa19(nama, nim, kelas, ipk));

        }

    }

}
```

6.5 Latihan Praktikum

6.5.1 Kode Program

- Kode Program DataDosen19

```
package Pertemuan6;

public class DataDosen19 {

    Dosen19 [] dataDosen = new Dosen19 [10];

    int idx;

    void tambah (Dosen19 m){

        if (idx<dataDosen.length){

            dataDosen[idx]=m;

            idx++;

        }else {

            System.out.println("data sudah penuh");

        }

    }

    void tampil (){

        for (int i = 0; i < idx; i++) {

            dataDosen[i].tampil();

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

        }

    }

    void bubbleSort(){

        for (int i = 0; i < idx - 1; i++){

            for (int j = 1; j < idx - i; j++){

                if (dataDosen[j] != null && dataDosen[j - 1] != null) {

                    if (dataDosen[j].usia < dataDosen[j - 1].usia){

                        Dosen19 tmp = dataDosen[j];

                        dataDosen[j] = dataDosen[j - 1];

                        dataDosen[j - 1] = tmp;

                    }

                }

            }

        }

    }

}
```

```

void selectionSort(){
    for (int i = 0; i < idx - 1; i++) {
        int idxMax = i;
        for (int j = i + 1; j < idx; j++) {
            if (dataDosen[j] != null && dataDosen[idxMax] != null) {
                if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                    idxMax = j;
                }
            }
        }
        if (idxMax != i) {
            Dosen19 tmp = dataDosen[idxMax];
            dataDosen[idxMax] = dataDosen[i];
            dataDosen[i] = tmp;
        }
    }
}

void insertionSort() {
    for (int i = 1; i < idx; i++) {
        Dosen19 temp = dataDosen[i];
        int j = i;
        if (temp != null) {
            while (j > 0 && dataDosen[j - 1] != null && dataDosen[j - 1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j - 1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}

```

```

void pencarianDataSequentialSearching19(String cari){
    int posisi = -1;
    int jumlahDitemukan = 0;
    for (int j=0; j<idx; j++){
        if (dataDosen[j] != null &&
dataDosen[j].nama.equalsIgnoreCase(cari)){
            if(posisi == -1){
                posisi=j;
            }
            jumlahDitemukan ++;
            tampilPosisi(cari, j);
            tampilDataSearch(cari, j);
        }
    }
    if (posisi == -1) {
        System.out.println("Data dosen dengan nama '" + cari + "' tidak
ditemukan.");
    } else if (jumlahDitemukan > 1) {
        System.out.println("\nPeringatan: Ditemukan lebih dari satu dosen
dengan nama '" + cari + "'!");
    }
}

void tampilPosisi(String x, int pos){
    System.out.println("Data dosen dengan nama "+x+" ditemukan pada indeks
"+pos);
}

void tampilDataSearch(String x, int pos){
    System.out.println("Nama\t : "+x);
    System.out.println("Kode\t : "+dataDosen[pos].kode);
    System.out.println("Jenis Kelamin\t : "+(dataDosen[pos].jenisKelamin ?
"Laki-laki":"Perempuan") );
    System.out.println("Usia\t : "+dataDosen[pos].usia);
}

```

```

    void pencarianDataBinary19(double cari, int left, int right, int
jmlDitemukan){

        if (right < left) {
            if (jmlDitemukan == 0) {
                System.out.println("Data dosen dengan usia '" + cari + "' tidak
ditemukan.");
            } else if (jmlDitemukan > 1) {
                System.out.println("Peringatan: Ditemukan lebih dari satu dosen
dengan usia " + cari + "!");
            }
            return;
        }

        int mid = (left + right) / 2;
        if (dataDosen[mid].usia == cari) {
            jmlDitemukan++;
            tampilPosisi2(cari, mid);
            tampilDataSearch2(mid);

            // Mengecek kiri dan kanan apakah ada data yang sama
            pencarianDataBinary19(cari, left, mid - 1, jmlDitemukan);
            pencarianDataBinary19(cari, mid + 1, right, jmlDitemukan);
        } else if (dataDosen[mid].usia < cari) {
            pencarianDataBinary19(cari, left, mid - 1, jmlDitemukan);
        } else {
            pencarianDataBinary19(cari, mid + 1, right, jmlDitemukan);
        }
    }

    void tampilPosisi2(double x, int pos) {
        System.out.println("Data dosen dengan usia " + x + " ditemukan pada
indeks " + pos);
    }

    void tampilDataSearch2( int pos) {
        System.out.println("Usia\t : " + dataDosen[pos].usia);
        System.out.println("Nama\t : " + dataDosen[pos].nama);
        System.out.println("Kode\t : " + dataDosen[pos].kode);

        System.out.println("Jenis Kelamin\t : " + (dataDosen[pos].jenisKelamin ?
"Laki-laki" : "Perempuan"));
    }
}

```

- Kode Program DosenMain19

```
package Pertemuan6;

import java.util.Scanner;

public class DosenMain19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        DataDosen19 data = new DataDosen19();

        while (true) {

            System.out.println("\n===== MENU DATA DOSEN =====");

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

            System.out.println("2. Tampilkan Data Dosen Sebelum Disorting");

            System.out.println("3. Sorting (Bubble Sort - Ascending Usia)");

            System.out.println("4. Sorting (Selection Sort - Descending Usia)");

            System.out.println("5. Sorting (Insertion Sort - Decending Usia)");

            System.out.println("6. Search Nama Menggunakan Sequential Search");

            System.out.println("7. Search Usia Menggunakan Binary Search");

            System.out.println("8. Keluar");

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

            int pilihan = sc.nextInt();

            sc.nextLine();

            if (pilihan == 1) {

                System.out.println("\nMasukkan Data Dosen");

                System.out.print("Kode Dosen: ");

                String kode = sc.nextLine();

                System.out.print("Nama Dosen: ");

                String nama = sc.nextLine();

                System.out.print("Jenis Kelamin (L/P): ");

                String kelamin = sc.nextLine();

                boolean jenisKelamin = kelamin.equalsIgnoreCase("L");

                System.out.print("Usia Dosen: ");

                int usia = sc.nextInt();

                sc.nextLine();

                data.tambah(new Dosen19(kode, nama, jenisKelamin, usia));

                System.out.println("Data dosen berhasil ditambahkan!");

            }

        }

    }

}
```

```

        } else if (pilihan == 2) {
            System.out.println("\n===== DATA DOSEN SEBELUM DISORTING
=====");

            data.tampil();
        } else if (pilihan == 3) {
            System.out.println("\nSorting menggunakan Bubble Sort ");
            data.bubbleSort();
            data.tampil();
        } else if (pilihan == 4) {
            System.out.println("\nSorting menggunakan Selection Sort ");
            data.selectionSort();
            data.tampil();
        } else if (pilihan == 5) {
            System.out.println("\nSorting menggunakan Insertion Sort ");
            data.insertionSort();
            data.tampil();
        } else if (pilihan == 6){
            System.out.println("-----");
            System.out.println("Pencarian data");
            System.out.println("-----");
            System.out.println("Masukkan nama dosen yang dicari: ");
            System.out.print("Nama: ");
            String cari = sc.nextLine();
            System.out.println("Menggunakan sequential searching");
            data.pencarianDataSequentialSearching19(cari);

```



```

        } else if (pilihan == 7){
            data.selectionSort();
            System.out.println("-----");
            System.out.println("Pencarian data");
            System.out.println("-----");
            System.out.println("Masukkan Usia Dosen yang dicari: ");
            System.out.print("Usia: ");
            double cari = sc.nextDouble();
            System.out.println("-----");
            System.out.println("Menggunakan binary search");
            System.out.println("-----");
            data.pencarianDataBinary19(cari, 0, data.idx-1, 0);
        } else if (pilihan == 8) {
            System.out.println("Program Selesai. Terima Kasih");
            break;
        } else {
            System.out.println("Pilihan tidak valid! Silakan pilih
kembali.");
        }
    }
}
}

```

6.5.2 Hasil

```
===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Search Nama Menggunakan Sequential Search
7. Search Usia Menggunakan Binary Search
8. Keluar
Pilih Menu: 1

Masukkan Data Dosen
Kode Dosen: 1
Nama Dosen: nuril
Jenis Kelamin (L/P): L
Usia Dosen: 20
Data dosen berhasil ditambahkan!

===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Search Nama Menggunakan Sequential Search
7. Search Usia Menggunakan Binary Search
8. Keluar
Pilih Menu: 1

Masukkan Data Dosen
Kode Dosen: 2
Nama Dosen: huda
Jenis Kelamin (L/P): P
Usia Dosen: 20
Data dosen berhasil ditambahkan!
```

===== MENU DATA DOSEN =====

1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Search Nama Menggunakan Sequential Search
7. Search Usia Menggunakan Binary Search
8. Keluar

Pilih Menu: 1

Masukkan Data Dosen

Kode Dosen: 3

Nama Dosen: nuril

Jenis Kelamin (L/P): L

Usia Dosen: 30

Data dosen berhasil ditambahkan!

===== MENU DATA DOSEN =====

1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Search Nama Menggunakan Sequential Search
7. Search Usia Menggunakan Binary Search
8. Keluar

Pilih Menu: 6

```
-----  
Pencarian data  
-----  
Masukkan nama dosen yang dicari:  
Nama: nuril  
Menggunakan sequential searching  
Data dosen dengan nama nuril ditemukan pada indeks 0  
Nama      : nuril  
Kode      : 1  
Jenis Kelamin : Laki-laki  
Usia      : 20  
Data dosen dengan nama nuril ditemukan pada indeks 2  
Nama      : nuril  
Kode      : 3  
Jenis Kelamin : Laki-laki  
Usia      : 30  
  
Peringatan: Ditemukan lebih dari satu dosen dengan nama 'nuril'!  
  
===== MENU DATA DOSEN =====  
1. Tambah Data Dosen  
2. Tampilkan Data Dosen Sebelum Disorting  
3. Sorting (Bubble Sort - Ascending Usia)  
4. Sorting (Selection Sort - Descending Usia)  
5. Sorting (Insertion Sort - Decending Usia)  
6. Search Nama Menggunakan Sequential Search  
7. Search Usia Menggunakan Binary Search  
8. Keluar  
Pilih Menu: 7
```

```

-----
Pencarian data
-----
Masukkan Usia Dosen yang dicari:
Usia: 20
-----
Menggunakan binary search
-----
Data dosen dengan usia 20.0 ditemukan pada indeks 1
Usia      : 20
Nama      : huda
Kode      : 2
Jenis Kelamin : Perempuan
Data dosen dengan usia 20.0 ditemukan pada indeks 2
Usia      : 20
Nama      : nuril
Kode      : 1
Jenis Kelamin : Laki-laki
Peringatan: Ditemukan lebih dari satu dosen dengan usia 20.0!
Peringatan: Ditemukan lebih dari satu dosen dengan usia 20.0!

===== MENU DATA DOSEN =====
1. Tambah Data Dosen
2. Tampilkan Data Dosen Sebelum Disorting
3. Sorting (Bubble Sort - Ascending Usia)
4. Sorting (Selection Sort - Descending Usia)
5. Sorting (Insertion Sort - Decending Usia)
6. Search Nama Menggunakan Sequential Search
7. Search Usia Menggunakan Binary Search
8. Keluar
Pilih Menu: 8
Program Selesai. Terima Kasih

```

Link Github : <https://github.com/nurilhuda05/Algoritma-dan-Struktur-Data/tree/aaffdb5741a3e7f0a162e2c9e10e7331886fa1aa/Pertemuan6>