

POLITEKNIK NEGERI MALANG
TEKNOLOGI INFORMASI
TEKNIK INFORMATIKA



Nama: Abdul Rahman Hanif Darmawan

NIM: 244107020232

Kelas: TI-1A

Prodi: D4-TEKNIK INFORMATIKA

JOBSHEET VI

SEARCHING

6.2 Searching / Pencarian Menggunakan Algoritma Sequential Search

6.2.2 Verifikasi Hasil Percobaan

```
Masukkan jumlah mahasiswa: 5
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 : 3,2
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
Kelas: 2
IPK: 3.1
-----
Nama: ila
NIM: 222
Kelas: 2
IPK: 3.2
-----
Nama: lia
NIM: 333
Kelas: 2
IPK: 3.2
-----
Nama: susi
NIM: 444
Kelas: 2
IPK: 3.5
-----
Nama: anita
NIM: 555
Kelas: 2
IPK: 3.7
-----
-----
Pencarian data
-----
masukkan ipk mahasiswa yang akan dicari :
IPK: 3,7
menggunakan sequential searching
data mahasiswa dengan IPK : 4.0 ditemukan pada indeks 4
nim : 555
nama : anita
kelas : 2
ipk : 3.7
```

6.2.3 Pertanyaan

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!
 - tampilPosisi hanya menampilkan indeks dimana IPK ditemukan
 - tampilDataSearch menampilkan detail lengkap mahasiswa seperti nim, nama, kelas, ipk.
2. Jelaskan fungsi break pada kode program dibawah ini!

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

- Menghentikan pencarian setelah menemukan ipk yang dicari, sehingga lebih cepat.

6.3 Searching / Pencarian Menggunakan Binary Search

6.3.2 Verifikasi Hasil Percobaan

```
Masukkan jumlah mahasiswa: 5  
Masukkan data mahasiswa ke-1  
NIM : 111  
Nama : adi  
Kelas : 2  
IPK : 3,1  
Masukkan data mahasiswa ke-2  
NIM : 222  
Nama : lia  
Kelas : 2  
IPK : 3,2  
Masukkan data mahasiswa ke-3  
NIM : 333  
Nama : lia  
Kelas : 2  
IPK : 3,3  
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  
Kelas: 2  
IPK: 3.1  
-----  
Nama: lia  
NIM: 222  
Kelas: 2  
IPK: 3.2  
-----  
Nama: lia  
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 search  
-----  
data mahasiswa dengan IPK : 3.7 ditemukan pada indeks 4  
nim : 555  
nama : anita  
kelas : 2  
ipk : 3.7
```

6.3.3 Pertanyaan

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

-

```
if (right >= left){  
    mid = (left+right)/2;  
    if (cari == listMhs[mid].ipk){
```

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!

- Tidak, karena Binary Search hanya berfungsi dengan data yang sudah diurutkan. Jika data tidak urut, pencarian bisa menghasilkan hasil yang salah atau tidak menemukan data yang seharusnya ada.

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

- Tidak, karena Binary Search umumnya bekerja pada data yang diurutkan secara ascending. Jika diubah hanya mengganti dari ">" menjadi "<"

```
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.

-

```
Scanner input = new Scanner(System.in);  
System.out.print("Masukkan jumlah mahasiswa: ");  
int jumlah = input.nextInt();  
input.nextLine();
```

6.5 Latihan Praktikum

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.
 - b) PencarianDataBinary : 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!

DataDosen02

```
package Tugas;
public class DataDosen02 {
    Dosen02[] dataDosen02 = new Dosen02[10];
    int idx = 0;

    public void tambah(Dosen02 dsn) {
        if (idx < dataDosen02.length) {
            dataDosen02[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data dosen sudah penuh!");
        }
    }

    public void tampil() {
        for (int i = 0; i < idx; i++) {
            dataDosen02[i].tampil();
        }
    }

    public void SortingASC() {}
    public void sortingDSC() {}
    public void insertionSort() {}

    public void PencarianDataSequential(String nama) {
        boolean ditemukan = false;
        int count = 0;
        for (int i = 0; i < idx; i++) {
            if (dataDosen02[i].nama.equalsIgnoreCase(nama)) {
                dataDosen02[i].tampil();
                ditemukan = true;
                count++;
            }
        }
        if (!ditemukan) {
            System.out.println("Data dosen dengan nama " + nama + " tidak ditemukan.");
        } else if (count > 1) {
            System.out.println("Peringatan: Ditemukan lebih dari satu hasil dengan nama yang sama!");
        }
    }

    public int PencarianDataBinary(int cari, int left, int right) {
        if (right >= left) {
            int mid = left + (right - left) / 2;

            if (dataDosen02[mid].usia == cari) {
                return mid;
            }
            if (dataDosen02[mid].usia > cari) {
                return PencarianDataBinary(cari, left, mid - 1);
            }
            return PencarianDataBinary(cari, mid + 1, right);
        }
        return -1;
    }
}
```

DosenMain02

```
package Tugas;
import java.util.Scanner;

public class DosenMain02 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen02 listDosen = new DataDosen02();
        boolean running = true;

        while (running) {
            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 Muda ke Tua)");
            System.out.println("4. Sorting DSC (Usia Tua ke Muda)");
            System.out.println("5. Pencarian Nama (Sequential
Search)");
            System.out.println("6. Pencarian Usia (Binary Search)");
            System.out.println("7. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            if (pilihan == 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 (Laki-laki
= true, Perempuan = false): ");
                boolean jk = sc.nextBoolean();
                System.out.print("Masukkan Usia Dosen: ");
                int usia = sc.nextInt();

                Dosen02 dsn = new Dosen02(kode, nama, jk, usia);
                listDosen.tambah(dsn);
            } else if (pilihan == 2) {
                listDosen.tampil();
            } else if (pilihan == 3) {
                listDosen.SortingASC();
                listDosen.tampil();
            } else if (pilihan == 4) {
                listDosen.sortingDSC();
                listDosen.tampil();
            } else if (pilihan == 5) {
                System.out.print("Masukkan Nama Dosen yang dicari:
");
                String cariNama = sc.nextLine();
                listDosen.PencarianDataSequential(cariNama);
            } else if (pilihan == 6) {

                System.out.print("Masukkan Usia Dosen yang dicari:
");
                int cariUsia = sc.nextInt();

                listDosen.SortingASC(); // Urutkan data sebelum
binary search
                int hasil = listDosen.PencarianDataBinary(cariUsia,
0, listDosen.idx - 1);

                if (hasil != -1) {
                    System.out.println("Data ditemukan.");
                    listDosen.dataDosen02[hasil].tampil();
                } else {
                    System.out.println("Data tidak ditemukan.");
                }
            } else if (pilihan == 7) {
                System.out.println("Program selesai.");
                running = false;
            } else {
                System.out.println("Pilihan tidak valid, silakan
coba lagi.");
            }
        }
    }
}
```

Output

```
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Pencarian Nama (Sequential Search)
6. Pencarian Usia (Binary Search)
7. Keluar
Pilih menu: 5
Masukkan Nama Dosen yang dicari: adi
Kode: 333
Nama: adi
Jenis Kelamin: Laki-laki
Usia: 26
-----
```

```
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Pencarian Nama (Sequential Search)
6. Pencarian Usia (Binary Search)
7. Keluar
Pilih menu: 6
Masukkan Usia Dosen yang dicari: 48
Data ditemukan:
Kode: witan
Nama: witan
Jenis Kelamin: Laki-laki
Usia: 48
-----
```

```
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Pencarian Nama (Sequential Search)
6. Pencarian Usia (Binary Search)
7. Keluar
Pilih menu: █
```

Link Github :

<https://github.com/baynobu/ALSD/tree/d4ccda3f138152ea040b0d902a85cec91f8a5d9b/Pertemuan%207>