

Nama : Siti Mutmainah

Kelas : TI-1B

Absen : 21

Mata Kuliah : ALSD

Langkah-langkah Percobaan Sequential Search :

1. Menggunakan code pada class jobsheet 6. Mahasiswa21, MahasiswaBerprestasi21, dan MahasiswaDemo21.
2. Tambahkan method sequentialSearching bertipe integer dengan parameter cari bertipe double pada class MahasiswaBerprestasi21. Kemudian Deklarasikan isi method sequentialSearching dengan algoritma pencarian data menggunakan teknik sequential searching.

```
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;  
}
```

3. Buatlah method tampilPoisisi bertipe void dan Deklarasikan isi dari method tampilPoisisi pada class MahasiswaBerprestasi21.

```
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");  
    }  
}
```

4. Pada class MahasiswaBerprestasi21, buatlah method tampilDataSearch bertipe void dan Deklarasikan isi dari method tampilDataSearch

```

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");
    }
}

```

5. Pada class MahasiswaDemo21, tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.

```

int jumMhs=5;

```

```

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

```

```

}
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:"menggunakan sequential searching");
double posisi = list.sequentialSearching(cari);
int pss= (int)posisi;
list.tampilPosisi(cari, pss);
list.tampilDataSearch(cari, pss);
}

```

6. Run dan hasil percobaan

```

PS D:\ASD\Praktikum-ASD\Jobsheet7> cd "d:\ASD\Praktikum-ASD\Jobsheet7\" ; if ($?) {
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 ditemukan pada indeks 3
nim      : 444
nama     : lia
kelas    : 2
ipk      : 3.5

```

Pertanyaan

1. Jelaskan perbedaan metod **tampilDataSearch** dan **tampilPosisi** pada class

MahasiswaBerprestasi!

Jawab: tampilPosisi hanya menampilkan posisi indeks dari data mahasiswa yang dicari berdasarkan ipk, tampilDataSearch menampilkan semua data mahasiswa yang ditemukan (NIM, Nama, Kelas, IPK)

2. Jelaskan fungsi **break** pada kode program dibawah ini!

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

Jawab: Fungsi break untuk menghentikan perulangan jika data yang dicari sudah ketemu, dan disimpan posisi indeks di variable posisi

Langkah-langkah Percobaan Binary Search

1. Pada percobaan sequential search tambahkan method findBinarySearch bertipe integer pada class MahasiswaBerprestasi. Kemudian Deklarasikan isi method findBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching

```
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. Panggil method findBinarySearchter dapat pada class MahasiswaBerprestasi di kelas MahasiswaDemo. Kemudia panggil method tampilPosisi dan tampilDataSearch

```
// Melakukan pencarian data Binary
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: ");
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, jumMhs - 1);
int pss2 = (int) posisi2;
list.tampilPosisi(cari, pss2);
list.tampilDataSearch(cari, pss2);
```

3. Run dan hasil percobaan

<pre>PS D:\ASD\Praktikum-ASD\Jobsheet7> cd "d:\ASD\Praktikum-ASD\Jobsheet7" iswaDemo21.java } ; if (\$?) { java Mahasiswa 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 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</pre>	<pre>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: 3.3 ----- Nama: susi NIM: 444 Kelas: 2 IPK: 3.5 ----- Nama: anita NIM: 555 Kelas: 2 IPK: 3.7 -----</pre>
---	--

```

-----
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

```

Pertanyaan

1. Tunjukkan kode program proses divide dijalankan

Jawab:

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

2. Tunjukkan kode program proses conquer dijalankan

Jawab:

```
if (cari == listMhs[mid].ipk){
    return mid;
}
```

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

Jawab: Program masih bisa berjalan tetapi hasil binary search akan salah, k karena program ini akan berjalan jika data sudah urut

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: Hail kemungkinan tidak ditemukan padahal nilai ada di data, karena binary search yang dipakai cocok untuk data dari kecil ke besar, sedangkan dalam kasus ini dari besar ke kecil

```
else if (listMhs[mid].ipk < cari){
```

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

Jawab:

```

public class MahasiswaBerprestasi21 {
    Run | Debug
    public static void main(String[] args){}
        Mahasiswa21[] listMhs;
        int idx = 0;

        public MahasiswaBerprestasi21(int jumlah) {
            listMhs = new Mahasiswa21[jumlah];
        }

```

```

import java.util.Scanner;
public class MahasiswaDemo21 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Masukkan jumlah mahasiswa yang ingin dimasukkan: ");
        int jumMhs = sc.nextInt();
        sc.nextLine();

        MahasiswaBerprestasi21 list = new MahasiswaBerprestasi21(jumMhs);

```

Ipk : 3.7

- PS D:\ASD\Praktikum-ASD\Jobsheet7> cd "d:\ASD\Praktikum-ASD\Jobsheet7"

```

Masukkan jumlah mahasiswa yang ingin dimasukkan: 3
Masukkan data mahasiswa ke-1

```