Nama : Dicky Darmawan
Kelas : TI 1B
NIM : 244107020037

Langkah-langkah Percobaan Sequential Search

1. Pada pertemuan Jobsheet 7 ini akan menggunakan class Mahasiswa<no Presensi>, MahasiswaBerprestasi<no Presensi>, dan MahasiswaDemo<no presensi> pada pertemuan Jobsheet 6 sebelumnya
2. Buat folder baru bernama Jobsheet7 di dalam repository Praktikum ASD, kemudian buka ketiga class dari Jobsheet 6 tersebut dan copy ke folder Jobsheet 7
3. Tambahkan method sequentialSearching bertipe integer dengan parameter cari bertipe double pada class MahasiswaBerprestasi<no presensi>. 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;
}
```

4. Buatlah method tampilPoisisi bertipe void dan Deklarasikan isi dari method tampilPoisisi pada class MahasiswaBerprestasi<no presensi>.

```
void tampilPosisi(double x, int pos) {
    if (pos != -1) {
        System.out.println("Data mahasiswa dengan
    } else {
        System.out.println("Data " + x + " tidak d
    }
}
```

5. Pada class MahasiswaBerprestasi<no presensi>, 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]
        System.out.println("Nama\t : " + listMhs[pos
        System.out.println("Kelas\t : " + listMhs[po
        System.out.println("IPK\t : " + listMhs[pos]
    } else {
        System.out.println("Data mahasiswa dengan IP
    }
}
```

6. Pada class MahasiswaDemo<noPresensi> , tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.

```
list.tampil();
System.out.println(x:"-------------------------------
System.out.println(x:"Pencarian data");
System.out.println(x:"-------------------------------
System.out.println(x:"Masukkan IPK mahasiswa yang dica
System.out.print(s:"IPK: ");
double cari = input.nextDouble();
System.out.println(x:"-------------------------");
System.out.println(x:"Menggunakan binary search");
System.out.println(x:"-------------------------");
double posisi2 = list.findBinarySearch(cari, left:0, j
int pss2 = (int) posisi2;
list.tampilPosisi(cari, pss2);
list.tampilDataSearch(cari, pss2);
```

7. Jalankan dan amati hasilnya.

Verifikasi Hasil Percobaan

```
--------------------------
Nama: Dicky
NIM: 111
Kelas: 1B
IPK: 3.8
--------------------------
Nama: Amel
NIM: 222
Kelas: 1B
IPK: 3.75
--------------------------
Nama: Arka
NIM: 333
Kelas: 1B
IPK: 3.6
--------------------------
Nama: Badar
NIM: 444
Kelas: 1B
IPK: 3.4
--------------------------
Nama: Putra
NIM: 555
Kelas: 1B
IPK: 3.5
--------------------------
---------------------------------
Pencarian data
---------------------------------
Masukkan IPK mahasiswa yang dicari:
IPK: 3,8
Menggunakan sequential searching
Data mahasiswa dengan IPK : 3.8 ditemukan pada indeks 0
NIM      : 111
Nama     : Dicky
Kelas    : 1B
IPK      : 3.8
---------------------------------
```

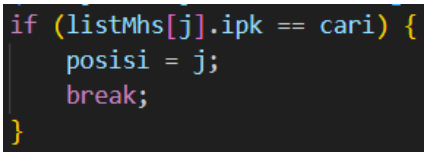
Pertanyaan

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

Method tampilPosisi digunakan jika hanya ingin mengetahui apakah data ditemukan dan di indeks berapa.

Method tampilDataSearch digunakan jika ingin melihat detail mahasiswa yang memiliki IPK tertentu.

2. Jelaskan fungsi break pada kode program dibawah ini!



Fungsi dari break dalam kode tersebut adalah menghentikan perulangan (loop) secara langsung ketika kondisi if terpenuhi.

Jika kondisi if (listMhs[j].ipk == cari) benar: maka variabel posisi diisi dengan nilai indeks j tempat IPK ditemukan

Kemudian break; menghentikan loop, sehingga pencarian berhenti ketika data pertama yang cocok ditemukan

Langkah-langkah Percobaan Binary Search

1. Pada percobaan 6.2.1 (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 righ
    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
        } else {
            return findBinarySearch(cari, mid + 1, r
        }
    }
    return -1;
}
```

2. Panggil method findBinarySearch terdapat pada class MahasiswaBerprestasi di kelas MahasiswaDemo. Kemudia panggil method tampilPosisi dan tampilDataSearch

```
list.tampil();
System.out.println(x:"------------------------------
System.out.println(x:"Pencarian data");
System.out.println(x:"------------------------------
System.out.println(x:"Masukkan IPK mahasiswa yang dica
System.out.print(s:"IPK: ");
double cari = input.nextDouble();
System.out.println(x:"-------------------------");
System.out.println(x:"Menggunakan binary search");
System.out.println(x:"-------------------------");
double posisi2 = list.findBinarySearch(cari, left:0, j
int pss2 = (int) posisi2;
list.tampilPosisi(cari, pss2);
list.tampilDataSearch(cari, pss2);
```

3. Jalankan dan amati hasilnya (inputkan data IPK secara terurut -ASC seperti verifikasi hasil percobaan dibawah ini).

Verifikasi Hasil Percobaan

```
-------------------------
Nama: Dicky
NIM: 111
Kelas: 1B
IPK: 3.8
-------------------------
Nama: Abel
NIM: 222
Kelas: 1B
IPK: 3.7
-------------------------
Nama: Bagas
NIM: 333
Kelas: 1B
IPK: 3.6
-------------------------
Nama: Hurin
NIM: 444
Kelas: 1B
IPK: 3.5
-------------------------
Nama: Julian
NIM: 555
Kelas: 1B
IPK: 3.5
-------------------------
-------------------------------------
Pencarian data
-------------------------------------
Masukkan IPK mahasiswa yang dicari:
IPK: 3,6
-------------------------
Menggunakan binary search
-------------------------
Data mahasiswa dengan IPK : 3.6 ditemukan pada indeks 2
NIM      : 333
Nama     : Bagas
Kelas    : 1B
IPK      : 3.6
```

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!

   Program masih dapat berjalan tetapi hasilnya tidak akan benar jika menggunakan binary search. Hal ini karena binary search hanya berfungsi jika data sudah dalam keadaan

terurut. Jika data tidak urut, maka pencarian bisa menghasilkan indeks yang salah atau bahkan gagal menemukan data yang ada

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

   binary search tidak akan berfungsi dengan benar, karena kode saat ini mengasumsikan data tersusun secara ascending.

```
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);
MahasiswaBerprestasi08 list = new MahasiswaBerprestas
System.out.println(x:"Masukkan jumlah mahasiswa: ");
int jumMhs = input.nextInt();

for (int i = 0; i < jumMhs; i++) {
    System.out.println("Masukkan data mahasiswa ke-"
```