

POLITEKNIK NEGERI MALANG
TEKNOLOGI INFORMASI
TEKNIK INFORMATIKA



Mohammad Ariq Baihaqi

244107020161

TI – 1A

16

6.2.1. Langkah-langkah Percobaan Sequential Search

```
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 denga 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: " + listMhs[pos].nim);
        System.out.println("Nama: " + listMhs[pos].nama);
        System.out.println("Kelas: " + listMhs[pos].kelas);
        System.out.println("IPK: " + listMhs[pos].ipk);
    } else {
        System.out.println("Data " + x + " tidak ditemukan");
    }
}
```

```
int jumMhs = 5;
```

```
for (int i = 0; i < jumMhs; i++) {  
    System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));  
    System.out.print("NIM: ");  
    String nim = sc.nextLine();  
    System.out.print("Nama: ");  
    String nama = sc.nextLine();  
    System.out.print("Kelas: ");  
    String kelas = sc.nextLine();  
    System.out.print("IPK: ");  
    double ipk = sc.nextDouble();  
    sc.nextLine();  
    System.out.println("-----");  
  
    list.tambah(new Mahasiswal6(nim, nama, kelas, ipk));  
}
```

```
list.tampilInformasi();
```

```
// Melakukan pencarian data mahasiswa berdasarkan sequential search
```

```
System.out.println("-----");  
System.out.println("Pencarian data");  
System.out.println("-----");  
System.out.print("Masukkan IPK mahasiswa yang dicari: ");  
double cari = sc.nextDouble();
```

```
System.out.println("Menggunakan sequential search");  
double posisi = list.sequentialSearching(cari);  
list.tampilPosisi(posisi, 0);  
list.tampilDataSearch(posisi, 0);
```

6.2.2. Verifikasi Hasil Percobaan

```
-----  
Nama: Fika  
NIM: 12321412  
Kelas: 2  
IPK: 3.0  
-----  
Nama: Ivana  
NIM: 3212321  
Kelas: 4  
IPK: 3.0  
-----  
Nama: Abid  
NIM: 213123123  
Kelas: 1  
IPK: 2.0  
-----  
-----  
Pencarian data  
-----  
Masukkan IPK mahasiswa yang dicari: 3  
Menggunakan sequential search  
data mahasiswa dengan IPK : 0.0 ditemukan pada indeks 0  
NIM: 244107020161  
Nama: Mohammad Ariq Baihaqi  
Kelas: 1  
IPK: 3.0  
Data sudah penuh  
Data sudah penuh  
Data sudah penuh  
Data sudah penuh  
Data sudah penuh  
Data Mahasiswa sebelum sorting:  
Nama: Mohammad Ariq Baihaqi  
NIM: 244107020161  
Kelas: 1  
IPK: 3.0  
-----
```

6.2.3. Pertanyaan

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

- Method tampilDataSearch digunakan menampilkan detail mahasiswa yang memiliki IPK sesuai dengan yang dicari
- Method tampilPosisi digunakan untuk posisi indeks mahasiswa yang ditemukan dalam array atau list

2. Jelaskan fungsi break pada kode program dibawah ini!

```
If (listMhs[j].ipk==cari) {  
    Posisi=j;  
    Break;  
}
```

- Mempercepat pencarian dengan menghentikan loop setelah menemukan hasil

6.3.1. Langkah-langkah Percobaan Binary Search

```
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;  
}
```

```
//melakukan pencarian data Binary

System.out.println("-----");
System.out.println("Pencarian data");
System.out.println("-----");
System.out.print("Masukkan IPK mahasiswa yang dicari: ");
System.out.print("IPK: ");
double cariBinary = sc.nextDouble();
System.out.println("-----");
System.out.println("Menggunakan Binary Search");
System.out.println("-----");
double posisi2 = list.findBinarySearch(cariBinary, jumMhs, jumMhs);
list.tampilPosisi(posisi2, jumMhs);
list.tampilDataSearch(posisi2, jumMhs);
```

6.3.2. Verifikasi Hasil Percobaan

Masukkan Data Mahasiswa ke-4

NIM: 121

Nama: EE

Kelas: 2

IPK: 4

Masukkan Data Mahasiswa ke-5

NIM: 1212

Nama: DDDD

Kelas: 3

IPK: 4

Nama: as

NIM: 233

Kelas: 2

IPK: 3.0

Nama: AB

NIM: 234

Kelas: 4

IPK: 4.0

Nama: l

NIM: 23

Kelas: D

IPK: 3.0

Nama: EE

NIM: 121

Kelas: 2

IPK: 4.0

Nama: DDDD

NIM: 1212

Kelas: 3

IPK: 4.0

Pencarian data

6.3.3. 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 tidakurut. Apakah program masih dapat berjalan? Mengapa demikian!**
 - Sequential search tetap bisa berjalan, tetapi binary search tidak akan berfungsi dengan benar
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**
 - Tidak sesuai, karena binary search mengasumsikan data dalam urutan menaik

```
int findBinarySearch(double cari, int left, int right, boolean descending) {
    if (right >= left) {
        int mid = (left + right) / 2;
        if (cari == listMhs[mid].ipk) {
            return mid;
        }
        if (descending) {
            // Jika data dalam urutan menurun
            if (listMhs[mid].ipk < cari) {
                return findBinarySearch(cari, left, mid - 1, true);
            } else {
                return findBinarySearch(cari, mid + 1, right, true);
            }
        } else {
            // Jika data dalam urutan naik
            if (listMhs[mid].ipk > cari) {
                return findBinarySearch(cari, left, mid - 1, false);
            } else {
                return findBinarySearch(cari, mid + 1, right, false);
            }
        }
    }
}
```



```
}  
  
}  
  
}  
  
return -1;  
  
}
```

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

```
System.out.print("Masukkan jumlah mahasiswa: ");  
    int jumMhs = sc.nextInt();  
  
    MahasiswaBerprestasi16modif list = new MahasiswaBerprestasi16modif();  
  
    for (int i = 0; i < jumMhs; i++) {  
        System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));  
        System.out.print("NIM: ");  
        String nim = sc.nextLine();  
        System.out.print("Nama: ");  
        String nama = sc.nextLine();  
        System.out.print("Kelas: ");  
        String kelas = sc.nextLine();  
        System.out.print("IPK: ");  
        double ipk = sc.nextDouble();  
        sc.nextLine();  
        System.out.println("-----");  
  
        list.tambah(new Mahasiswa16(nim, nama, kelas, ipk));  
    }  
  
    list.tampilInformasi();
```

```
// untuk menyesuaikan jumlah mahasiswa

public MahasiswaBerprestasi16modif(int kapasitas) {

    listMhs = new Mahasiswa16[kapasitas];

}

void tambah(MahasiswaDemol6modif m1) {

    if (idx < listMhs.length) {

        listMhs[idx] = m1;

        idx++;

    } else {

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

    }

}
```

OUTPUT

```
-----  
Masukkan Data Mahasiswa ke-3  
NIM: 233  
Nama: Fika  
Kelas: 3  
IPK: 3  
-----  
Nama: 233  
NIM:  
Kelas: 3  
IPK: 4.0  
-----  
Nama: Ariq  
NIM: 255  
Kelas: 4  
IPK: 4.0  
-----  
Nama: Fika  
NIM: 233  
Kelas: 3  
IPK: 3.0  
-----  
Data sudah penuh  
Data sudah penuh  
Data sudah penuh  
Data sudah penuh  
Data sudah penuh  
Data Mahasiswa sebelum sorting:  
Nama: 233  
NIM:  
Kelas: 3  
IPK: 4.0  
-----  
Nama: Ariq  
NIM: 255  
Kelas: 4  
IPK: 4.0  
-----
```

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!

```
// a. Sequential Search berdasarkan nama

public void pencarianDataSequential16(String namaCari) {
    int count = 0;
    System.out.println("Hasil pencarian nama \"" + namaCari + "\"");
    for (int i = 0; i < idx; i++) {
        if (dataDosen[i].nama.equalsIgnoreCase(namaCari)) {
            dataDosen[i].tampil();
            count++;
        }
    }
    if (count == 0) {
        System.out.println("Data tidak ditemukan.");
    } else if (count > 1) {
        System.out.println("Peringatan: Ditemukan lebih dari satu dosen dengan nama yang sama!");
    }
}

// b. Binary Search berdasarkan usia

public void pencarianDataBinary16(int usiaCari) {
    sortingDSC();
    int left = 0, right = idx - 1;
    int mid, count = 0;
```

```

while (left <= right) {
    mid = (left + right) / 2;
    if (dataDosen[mid].usia == usiaCari) {
        System.out.println("Hasil pencarian usia " + usiaCari + ":");
        dataDosen[mid].tampil();
        count++;

        // Cek apakah ada hasil lain dengan usia yang sama (di kiri &
kanan)

        int tempMid = mid;
        while (--tempMid >= 0 && dataDosen[tempMid].usia == usiaCari) {
            dataDosen[tempMid].tampil();
            count++;
        }
        tempMid = mid;
        while (++tempMid < idx && dataDosen[tempMid].usia == usiaCari) {
            dataDosen[tempMid].tampil();
            count++;
        }

        if (count > 1) {
            System.out.println("Peringatan: Ditemukan lebih dari satu dosen
dengan usia yang sama!");
        }
        return;
    } else if (dataDosen[mid].usia > usiaCari) {
        right = mid - 1;
    } else {
        left = mid + 1;
    }
}
}

```

OUTPUT

```
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort
4. Urutkan DSC (Usia Tua ke Muda) - Selection Sort
5. Keluar
Pilih menu: 3
Kode: 001, Nama: Ariq, Jenis Kelamin: Laki-laki, Usia: 25
Kode: 002, Nama: Fika, Jenis Kelamin: Perempuan, Usia: 20
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort
4. Urutkan DSC (Usia Tua ke Muda) - Selection Sort
5. Keluar
Pilih menu: 3
Data telah diurutkan ASC.
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort
4. Urutkan DSC (Usia Tua ke Muda) - Selection Sort
5. Keluar
Pilih menu: 3
Data telah diurutkan ASC.
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort
2. Tampilkan Data Dosen
3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort
4. Urutkan DSC (Usia Tua ke Muda) - Selection Sort
5. Keluar
Pilih menu: 3
Data telah diurutkan ASC.
Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
```

Link Github : <https://github.com/Ariqq16/semester2>

