

## Jobsheet 6

### Algoritma dan Struktur Data

Hernanda Rizka Utami / 244107060075 / SIB – 1B

6.2	<b>Searching (Sequential Search)</b>
	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <pre> Masukkan data mahasiswa ke-1: Nama : adi NIM : 111 Kelas : 2 IPK : 3,6 Masukkan data mahasiswa ke-2: Nama : tio NIM : 222 Kelas : 2 IPK : 3,8 Masukkan data mahasiswa ke-3: Nama : ila NIM : 333 Kelas : 2 IPK : 3 Masukkan data mahasiswa ke-4: Nama : lia NIM : 444 Kelas : 2 IPK : 3,5 Masukkan data mahasiswa ke-5: Nama : fia NIM : 555 Kelas : 2 IPK : 3,3 </pre> </div> <div style="width: 45%;"> <pre> NIM : 111 Nama : adi Kelas : 2 IPK : 3,6 ----- NIM : 222 Nama : tio Kelas : 2 IPK : 3,8 ----- NIM : 333 Nama : ila Kelas : 2 IPK : 3,0 ----- NIM : 444 Nama : lia Kelas : 2 IPK : 3,5 ----- NIM : 555 Nama : fia Kelas : 2 IPK : 3,3 ----- </pre> </div> </div> <div style="margin-top: 20px;"> <pre> ----- Pencaarian Data ----- masukkan ipk mahaiswa 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 </pre> </div>

6.2.3	Pertanyaan	Jawaban
1.	Jelaskan perbedaan metod <b>tampilDataSearch</b> dan <b>tampilPosisi</b> pada class MahasiswaBerprestasi!	tampilDataSearch itu menampilkan seluruh informasi mahasiswa yang ditemukan, sedangkan tampilPosisi itu untuk menunjukkan dimana data berada (indeksnya)
2.	Jelaskan fungsi <b>break</b> pada kode program dibawah ini! <pre> if (listMhs[j].ipk==cari){     posisi=j;     break; } </pre>	Break berfungsi untuk menghentikan proses perulangan saat kondisi terpenuhi.

Searching (Binary Search)	
6.3	<pre> Masukkan data mahasiswa ke-1: Nama : adi NIM : 111 Kelas : 2 IPK : 3,1 Masukkan data mahasiswa ke-2: Nama : ila NIM : 222 Kelas : 2 IPK : 3,2 Masukkan data mahasiswa ke-3: Nama : lia NIM : 333 Kelas : 2 IPK : 3,3 Masukkan data mahasiswa ke-4: Nama : susi NIM : 444 Kelas : 2 IPK : 3,5 Masukkan data mahasiswa ke-5: Nama : anita NIM : 555 Kelas : 2 IPK : 3,7 </pre> <pre> NIM : 111 Nama : adi Kelas : 2 IPK : 3.1 ----- NIM : 222 Nama : ila Kelas : 2 IPK : 3.2 ----- NIM : 333 Nama : lia Kelas : 2 IPK : 3.3 ----- NIM : 444 Nama : susi Kelas : 2 IPK : 3.5 ----- NIM : 555 Nama : anita Kelas : 2 IPK : 3.7 ----- </pre> <pre> ----- menggunakan binary search ----- data mahasiswa dengan ipk: 3.7 ditemukan pada indeks: 4 nim      :555 nama     :anita kelas    :2 ipk      :3.7 </pre>

6.3.3	Pertanyaan	Jawaban
1.	Tunjukkan pada kode program yang mana proses divide dijalankan!	<p>Proses divide adalah bagian yang membagi masalah menjadi sub-masalah yang lebih kecil. Pada program, divide terdapat pada method findBinarySearch yang pada sintaks <math>mid = (left + right) / 2</math> adalah bagian pembagian (divide), lalu membagi masalah berdasarkan perbandingan yaitu pada sintaks <code>return findBinarySearch(cari, left, mid-1)</code> jika elemen mid lebih besar dari yang dicari maka cari pembagian dikiri dan jika elemen mid lebih kecil dari yang dicari maka cari pembagian dikanan, proses akan berulang secara rekrusif hingga elemen ditemukan atau ruang pencarian habis.</p> <pre> mid = (left+right)/2; if(cari &lt; listMhs[mid].ipk){     return findBinarySearch(cari, left, mid-1); } return findBinarySearch(cari, mid+1, right); </pre>
2.	Tunjukkan pada kode program yang mana proses conquer dijalankan!	<p>Proses conquer terjadi ketika solusi dari sub-masalah yang lebih kecil digabungkan atau ketika elemen yang dicari ditemukan. Pada method findBinarySearch conquer terjadi dalam dua scenario, pertama pada saat data ditemukan, ketika nilai ipk yang dicari ditemukan persis pada posisi mid, maka program akan langsung mengembalikan posisi tersebut namun pada proses tersebut tidak ada penggabungan karna sub-problem langsung menjadi solusi problem utama. Kedua saat data tidak ditemukan, jika ruang pencarian sudah habis (<math>right &lt; left</math>) dan data tidak ditemukan program akan mengembalikan -1 sebagai solusi akhir. Dalam binary search karakteristik conquer yaitu tidak ada tahap penggabungan kompleks seperti merge sort, solusi sub-masalah langsung menjadi solusi akhir, setiap pemanggilan rekrusif akan mengembalikan solusi ke pemanggil sebelumnya.</p> <pre> if(cari == listMhs[mid].ipk){     return mid; } return -1; </pre>
3.	Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!	Tidak, program binary search tidak akan berjalan dengan benar jika data ipk tidak diurutkan terlebih dahulu karena binary search membutuhkan prasyarat data yang telah terurut untuk berfungsi secara akurat.
4.	Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (misal :	Hasil dari binary search tidak sesuai untuk data descending karena logika pencariannya dirancang untuk data ascending. Ketika mencari ipk 3,2 pada array yang disebutkan, algoritma akan salah mengarahkan pencarian ke bagian kiri (indeks lebih rendah) yang seharusnya ke kanan

	<p>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</p>	<p>(indeks lebih tinggi) sehingga mengembalikan -1 atau tidak ditemukan meskipun data ada. Solusinya dengan membalik operator perbandingan (&gt; menjadi &lt;) agar sesuai dengan urutan descending.</p> <pre data-bbox="560 325 1117 399"> else if(listMhs[mid].ipk &gt; cari){ </pre> <p>Menjadi</p> <pre data-bbox="560 430 1140 504"> else if(listMhs[mid].ipk &lt; cari){ </pre>
5.	<p>Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.</p>	<pre data-bbox="560 630 1388 913"> System.out.print("masukkan jumlah mahasiswa yang akan diinput: "); int jumMhs = sc.nextInt(); sc.nextLine();  mahasiswaBerprestasi12 list = new mahasiswaBerprestasi12(jumMhs);  for (int i = 0; i &lt; jumMhs; i++) {     System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");     System.out.print("Nama    : ");     String nama = sc.nextLine(); </pre>

6.4	Latihan Praktikum
1.	<p>Pada Latihan praktikum pertemuan sebelumnya pada jobsheet 5 yang terdapat 3 class yaitu <b>Dosen</b>&lt;no presensi&gt;, <b>DataDosen</b>&lt;no presensi&gt; , dan <b>DosenDemo</b>&lt;no presensi&gt;, tambahkan method:</p>
a.	<p>PencarianDataSequensial&lt;no presensi&gt; : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.</p> <pre data-bbox="297 432 1226 808"> public void pencarianDataSequensial(String nama){     boolean found = false;     for(int i =0; i&lt; idx; i++){         if(dataDosen[i].nama.equalsIgnoreCase(nama)){             System.out.println("Data dosen ditemukan: ");             dataDosen[i].tampil();             found = true;             break;         }     }     if(!found){         System.out.println("Data dosen dengan nama " + nama + " tidak ditemukan!");     } } </pre> <pre data-bbox="297 835 1226 1045"> break; case 6:     System.out.print("Masukkan nama dosen yang dicari: ");     String namaCari = sc.nextLine();     daftarDosen.pencarianDataSequensial(namaCari);     break; </pre> <pre data-bbox="297 1083 662 1633"> Pilih menu: 2 Kode      : 111 Nama      : lia Jenis Kelamin : Perempuan Usia      : 25 ----- Kode      : 333 Nama      : mashur Jenis Kelamin : Laki-laki Usia      : 30 ----- Kode      : 222 Nama      : tono Jenis Kelamin : Laki-laki Usia      : 34 ----- Kode      : 44 Nama      : anita Jenis Kelamin : Perempuan Usia      : 45 ----- Kode      : 555 Nama      : putri Jenis Kelamin : Perempuan Usia      : 29 ----- </pre> <pre data-bbox="670 1304 1287 1633"> 6. Cari Dosen berdasarkan Nama: 7. Keluar Pilih menu: 6 Masukkan nama dosen yang dicari: tono Data dosen ditemukan: Kode      : 222 Nama      : tono Jenis Kelamin : Laki-laki Usia      : 34 ----- </pre>

	<pre> MENU: 1. Tambah Data Dosen 2. Tampil Data Dosen 3. Sorting ASC (Bubble Sort) 4. Sorting DSC (Selection Sort) 5. Sorting DSC (Insertion Sort) 6. Cari Dosen berdasarkan Nama: 7. Keluar Pilih menu: 6 Masukkan nama dosen yang dicari: siti Data dosen dengan nama siti tidak ditemukan! </pre>
b.	<p>PencarianDataBinary&lt;no presensi&gt; : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.</p> <pre> public void pencarianDataBinary12(int usia){     SortingASC();     int left =0, right =idx-1;     boolean found = false;     while (left&lt;=right){         int mid = left + (right - left)/2;         if(dataDosen[mid].usia == usia){             System.out.println("Data dosen ditemukan: ");             dataDosen[mid].tampil();             found = true;             break;         }         else if(dataDosen[mid].usia &lt; usia){             left = mid+1;         }         else{             right = mid -1;         }     }     if(!found){         System.out.println("Data dosen dengan usia " + usia + " tidak ditemukan!");     } } </pre> <pre>         break;     case 7:         System.out.print("Masukkan usia dosen yang dicari: ");         int usiaCari = sc.nextInt();         daftarDosen.pencarianDataBinary12(usiaCari);         break; </pre>

```

Pilih menu: 2
Kode      : 112
Nama       : budi
Jenis Kelamin : Laki-laki
Usia       : 50
-----
Kode      : 221
Nama       : wilda
Jenis Kelamin : Perempuan
Usia       : 30
-----
Kode      : 332
Nama       : rachmad
Jenis Kelamin : Laki-laki
Usia       : 43
-----
Kode      : 443
Nama       : susi
Jenis Kelamin : Perempuan
Usia       : 36
-----
Kode      : 554
Nama       : titik
Jenis Kelamin : Perempuan
Usia       : 29
-----

```

```

7. Cari Dosen berdasarkan Usia:
8. Keluar
Pilih menu: 7
Masukkan usia dosen yang dicari: 30
Data dosen ditemukan:
Kode      : 221
Nama       : wilda
Jenis Kelamin : Perempuan
Usia       : 30
-----

```

```

7. Cari Dosen berdasarkan Usia:
8. Keluar
Pilih menu: 7
Masukkan usia dosen yang dicari: 20
Data dosen dengan usia 20 tidak ditemukan!

```

- c. Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

```

public void pencarianDataBinary12(int usia){
    SortingASC();
    int left = 0, right = idx-1;
    boolean found = false;
    int count = 0;

    while (left <= right){
        int mid = left + (right - left)/2;
        if(dataDosen[mid].usia == usia){
            System.out.println("Data dosen ditemukan: ");
            int i = mid;

            while (i >= 0 && dataDosen[i].usia == usia){
                dataDosen[i].tampil();
                count++;
                i--;
            }
            i = mid+1;
            while(i < idx && dataDosen[i].usia == usia){
                dataDosen[i].tampil();
                count++;
                i++;
            }
            found = true;
            break;
        }
    }
}

```

```

        break;
    }
    else if(dataDosen[mid].usia < usia){
        left = mid+1;
    }
    else{
        right = mid -1;
    }
}
if(!found){
    System.out.println("Data dosen dengan usia " + usia + " tidak ditemukan!");
}
else if(count > 1){
    System.out.println("Peringatan: Terdapat lebih dari satu data dosen\n dengan usia yang sama!");
}
}
}

```

```

8. Keluar
Pilih menu: 2
Kode       : 123
Nama       : agus
Jenis Kelamin : Laki-laki
Usia       : 35
-----
Kode       : 234
Nama       : suryo
Jenis Kelamin : Laki-laki
Usia       : 29
-----
Kode       : 345
Nama       : imam
Jenis Kelamin : Laki-laki
Usia       : 35
-----
Kode       : 456
Nama       : yunus
Jenis Kelamin : Laki-laki
Usia       : 55
-----
Kode       : 567
Nama       : husna
Jenis Kelamin : Perempuan
Usia       : 30
-----

```

```

8. Keluar
Pilih menu: 6
Masukkan nama dosen yang dicari: imam
Data dosen ditemukan:
Kode       : 345
Nama       : imam
Jenis Kelamin : Laki-laki
Usia       : 35
-----

```

```

8. Keluar
Pilih menu: 7
Masukkan usia dosen yang dicari: 35
Data dosen ditemukan:
Kode       : 123
Nama       : agus
Jenis Kelamin : Laki-laki
Usia       : 35
-----
Kode       : 345
Nama       : imam
Jenis Kelamin : Laki-laki
Usia       : 35
-----
Peringatan: Terdapat lebih dari satu data dosen
dengan usia yang sama!

```