

LAPORAN PRAKTIKUM ALGOTIRMA DAN STRUKTUR DATA

JOBSHEET 6 SEARCHING

Dosen Pengajar : Triana Fatmawati, S.T., M.T.



Muhammad Afiq Firdaus

2341760189 / 21

SIB-1E

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

6.2.1 Sequential Search Menggunakan Array

Berikut adalah hasil code berdasarkan jobsheet Algoritma Sequential Search

```
J Sorting21.java > Sorting21
1 public class Sorting21 {
2     public static void sequentialSearch(int[] arr, int key) {
3         for (int i = 0; i < arr.length; i++) {
4             if (i == arr[i]) {
5                 System.out.println("Data ditemukan pada indeks ke- " + i);
6             }
7         }
8         System.out.println(x:"Data tidak ditemukan");
9     }
10
11     Run | Debug
12     public static void main(String[] args) {
13         int[] daftarNilai = { 10, 5, 20, 15, 80, 45 };
14         sequentialSearch(daftarNilai, key:5);
15     }
}
```

Berikut adalah hasil run dari code diatas

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07> c::;
aktikum\Jobsheet 07'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+Sh
ode\User\workspaceStorage\a724fe32959bc91cd8f73755b55738c1\redhat.java\jdt_ws\Jobsheet 07_fa4f4
Data tidak ditemukan
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07> []
```

6.2.2 Sequential Search Menggunakan Array of Object

Berikut adalah code class Mahasiswa

```
J Mahasiswa21.java > ...
1 public class Mahasiswa21 {
2     int nim;
3     String nama;
4     int umur;
5     double ipk;
6
7
8     Mahasiswa21(int ni, String n, int u, double i){
9         nim = ni;
10        nama = n;
11        umur = u;
12        ipk = i;
13    }
14
15    void tampil(){
16        System.out.println("Nim = " + nim);
17        System.out.println("Nama = " + nama);
18        System.out.println("Umur = " + umur);
19        System.out.println("IPK = " + ipk);
20    }
21 }
```

Berikut adalah code class pencarianMhs

```
PencarianMhs21.java > PencarianMhs21
1
2 public void Tampilposisi(int x, int pos)
3 {
4     if(pos != -1){
5         System.out.println("data : " + x + "ditemukan pada indeks" + pos);
6     }else{
7         System.out.println("data" + x + "tidak ditemukan");
8     }
9
10    public void TampilData(int x, int pos)
11    {
12        if(pos != -1){
13            System.out.println("Nim\t : " + x);
14            System.out.println("Nama\t : " + listMhs[pos].nama);
15            System.out.println("Umur\t : " + listMhs[pos].umur);
16            System.out.println("IPK\t : " + listMhs[pos].ipk);
17
18            public void TampilData(int x, int pos)
19            {
20                if(pos != -1){
21                    System.out.println("Nim\t : " + x);
22                    System.out.println("Nama\t : " + listMhs[pos].nama);
23                    System.out.println("Umur\t : " + listMhs[pos].umur);
24                    System.out.println("IPK\t : " + listMhs[pos].ipk);
25                }else{
26                    System.out.println("Data " + x + "tidak ditemukan");
27                }
28            }
29        }
30    }
31 }
```

```
J PencarianMhs21.java > PencarianMhs21
1 // PERCOBAAN 1
2 public class PencarianMhs21 {
3     Mahasiswa21 listMhs[] = new Mahasiswa21[10];
4     int idx;
5
6     void tambah (Mahasiswa21 m){
7         if (idx < listMhs.length){
8             listMhs[idx] = m;
9             idx++;
10        }else{
11            System.out.println("Maksimum tercapai");
12        }
13    }
14
15    void tampil(){
16        for(Mahasiswa21 m : listMhs){
17            m.tampil();
18            System.out.println("-----");
19        }
20    }
21
22    public int FindSeqSearch (int x){
23        int posisi = -1;
24        for(int j=0; j< listMhs.length; j++){
25            if(listMhs[j].nim==x){
26                posisi = j;
27                break;
28            }
29        }
30        return posisi;
31    }
```

Berikut adalah code MahasiswaMain

```
Welcome | Sorting21.java | Mahasiswa21.java | PencarianMhs21.java | MahasiswaMain.java 2
MahasiswaMain.java > ...
1 // PERCOBAAN 1
2 import java.util.Scanner;
3
4 public class MahasiswaMain {
5     public static void main(String[] args) {
6         Scanner s = new Scanner(System.in);
7         Scanner sl = new Scanner(System.in);
8
9         PencarianMhs21 data = new PencarianMhs21();
10        int jumMhs = 5;
11
12        System.out.println(x:"-----");
13        System.out.println(x:"Masukkan data Mahasiswa secara Urut dari Nim terkecil:");
14        for (int i = 0; i < jumMhs; i++) {
15            System.out.println(x:"-----");
16            System.out.println(x:"Nim\t: ");
17            int nim = s.nextInt();
18            System.out.println(x:"Nama\t: ");
19            String nama = sl.nextLine();
20            System.out.println(x:"Umur\t: ");
21            int umur = s.nextInt();
22            System.out.println(x:"IPK\t: ");
23            double ipk = s.nextDouble();
```

```

24
25        Mahasiswa21 m = new Mahasiswa21(nim, nama, umur, ipk);
26        data.tambah(m);
27    }
28    System.out.println(x:"-----");
29    System.out.println(x:"Data Keseluruhan Mahasiswa: ");
30    data.tampil();
31
32    System.out.println(x:"-----");
33    System.out.println(x:"-----");
34    System.out.println(x:"Pencarian Data: ");
35    System.out.println(x:"Masukkan Nim Mahasiswa yang dicari: ");
36    System.out.println(x:"NIM : ");
37    int cari = s.nextInt();
38    System.out.println(x:"Menggunakan sequential serach");
39    int posisi = data.FindSeqSearch(cari);
40
41    data.Tampilposisi(cari, posisi);
42
43    data.TampilData(cari, posisi);
44 }
45 }
```


Dan berikut adalah hasil run dari code-code diatas

```
3,0
-----
Data Keseluruhan Mahasiswa:
Nim = 2017
Nama = Dewi Lestari
Umur = 23
IPK = 3.5
-----
Nim = 2018
Nama = Sinta Sanjaya
Umur = 22
IPK = 4.0
-----
Nim = 2019
Umur = 22
IPK = 3.7
-----
Nim = 2020
Nama = Budi Prakarsa
Umur = 20
IPK = 2.9
-----
Nim = 2021
Nama = Yania Siti
Umur = 20
IPK = 3.0
-----
```

```
-----
Pencarian Data:
Masukkan Nim Mahasiswa yang dicari:
NIM :
2018
Menggunakan sequential search
data : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan
aktikum\Jobsheet 07'; & 'C:\Program Files\Java\jdk-2
ode\User\workspaceStorage\a724fe32959bc91cd8f73755b5
```

Pertanyaan :

1. Lakukan perubahan array daftarNilai pada fungsi main()

```
Run | Debug
public static void main(String[] args) {
    int[] daftarNilai = { 10, 5, 20, 15, 5, 45 };
    sequentialSearch(daftarNilai, 5);
}
```

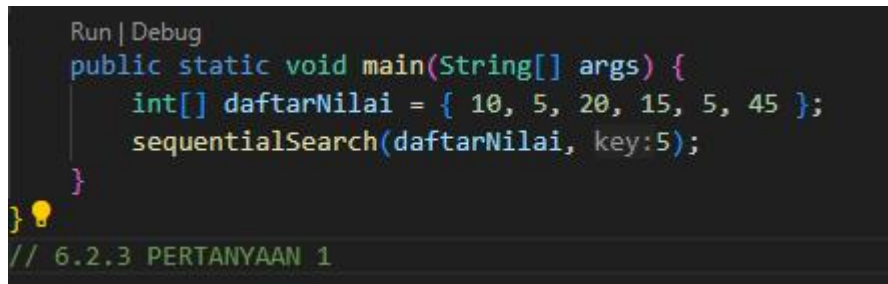
2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs
3. Jelaskan fungsi break pada kode program dibawah ini!

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

4. Jika Data Nim yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Mengapa demikian!

Jawaban :

1.

A screenshot of a code editor with a dark background. At the top, it says 'Run | Debug'. The code is in Java and defines a main method that takes a String array 'args'. Inside the main method, an integer array 'daftarNilai' is initialized with the values {10, 5, 20, 15, 5, 45}. Then, the 'sequentialSearch' method is called with 'daftarNilai' and 'key:5' as arguments. The code is enclosed in curly braces. At the bottom left, there is a yellow lightbulb icon. At the bottom, a comment '// 6.2.3 PERTANYAAN 1' is visible.

```
Run | Debug
public static void main(String[] args) {
    int[] daftarNilai = { 10, 5, 20, 15, 5, 45 };
    sequentialSearch(daftarNilai, key:5);
}
// 6.2.3 PERTANYAAN 1
```

2. perbedaan utama antara kedua metode tersebut adalah dalam informasi yang ditampilkan. Metode TampilData menampilkan seluruh informasi mahasiswa yang ditemukan, sedangkan metode Tampilposisi hanya menampilkan indeks posisi di mana data ditemukan.

3. Fungsi dari break; adalah untuk menghentikan iterasi loop secara paksa segera setelah suatu elemen yang memenuhi kondisi pencarian ditemukan. Ini menghindarkan loop dari melanjutkan iterasi ke elemen berikutnya setelah elemen yang sesuai dengan kriteria pencarian telah ditemukan, karena sudah tidak perlu lagi mencari. Dengan demikian, break; mengoptimalkan kinerja dan menghindari pemrosesan yang tidak perlu dalam pencarian.

4. Ya, program ini akan tetap berjalan meskipun data NIM yang dimasukkan tidak terurut dari kecil ke besar. Namun, hasil yang dikeluarkan mungkin tidak akan selalu benar.

Alasan program masih dapat berjalan adalah karena pencarian dilakukan secara sequential (berurutan) melalui seluruh elemen array listMhs untuk mencari NIM yang cocok dengan nilai yang dicari. Namun, jika data NIM tidak terurut, pencarian sequential ini mungkin memerlukan lebih banyak iterasi daripada jika data terurut, karena harus memeriksa setiap elemen dari awal hingga akhir array.

Ketika data tidak terurut, hasil dari pencarian tergantung pada posisi mana elemen yang sesuai ditemukan. Jika elemen yang sesuai berada di posisi awal array, hasilnya akan benar. Namun, jika elemen yang sesuai berada di posisi akhir array, hasilnya juga akan benar. Namun, jika elemen yang sesuai berada di tengah array atau tidak ada dalam array, hasilnya akan salah.

Jadi, meskipun program masih dapat berjalan, keakuratannya tergantung pada posisi elemen yang sesuai dalam array, dan jika data tidak terurut, hasilnya mungkin tidak selalu benar.

6.3. Searching / Pencarian Menggunakan Binary Search

6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

1. Tambahkan method `binarySearchAsc()` pada file `Sorting.java`

```
11 public static int binarySearchAsc(int[] arr, int key) {
12     int start = 0, end = arr.length - 1;
13
14     while (start <= end) {
15         int mid = start + (end - start) / 2;
16
17         if (arr[mid] == key) {
18             start = mid + 1;
19         } else {
20             end = mid - 1;
21         }
22     }
23     return -1;
24 }
```

2. Tambahkan baris program untuk menguji method `binarySearchAsc()` pada fungsi `main()`

```
Run | Debug
6 public static void main(String[] args) {
7     int[] daftarNilai = { 10, 5, 20, 15, 80, 45 };
8     sequentialSearch(daftarNilai, key:5);
9
10    int[] sortedNilai = { 5, 5, 10, 20, 30, 40, 50 };
11    int index = binarySearchAsc(sortedNilai, key:5);
12
13    if (index != -1) {
14        System.out.println("Data ditemukan pada indeks ke-" + index);
15    } else {
16        System.out.println("Data tidak ditemukan");
17    }
18 }
```

3. Run dan compile program

```
07'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInEx
959bc91cd8f73755b55738c1\redhat.java\jdt_ws\Jobsheet 07_fa4f4087\bin' 'Sorting21'
Data tidak ditemukan
Data tidak ditemukan
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07>
```

6.3.2. Langkah-langkah Percobaan Binary Search menggunakan Array of Object

1. Pada percobaan 6.2.2 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.

```
104 public int FindBinarySearch(int cari, int left, int right) {  
105     int mid;  
106     if (right >= left) {  
107         mid = (left + right) / 2;  
108         if (cari == listMhs[mid].nim) {  
109             return (mid);  
110         } else if (listMhs[mid].nim > cari) {  
111             return FindBinarySearch(cari, left, mid - 1);  
112         } else {  
113             return FindBinarySearch(cari, mid + 1, right);  
114         }  
115     }  
116     return -1;  
117 }  
118  
119 }
```

2. Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas Mahasiswamain. Kemudian panggil method tampilposisi dan tampilData

```
System.out.println(x:"=====");  
System.out.println(x:"Menggunakan binary Search");  
posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1);  
data.Tampilposisi(cari, posisi);  
data.TampilData(cari, posisi);  
}
```

3. Jalankan dan amati hasilnya.

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\aktikum\Jobsheet 07'; & 'C:\Program Files\Java\jdk-21\bin\java.exe'  
ode\User\workspaceStorage\a724fe32959bc91cd8f73755b55738c1\redhat.j  
Data tidak ditemukan  
Data tidak ditemukan  
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\
```


6.3.2. Langkah-langkah Percobaan Binary Search menggunakan Array of Object

1. Pada percobaan 6.2.2 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.

```
104     public int FindBinarySearch(int cari, int left, int right) {
105         int mid;
106         if (right >= left) {
107             mid = (left + right) / 2;
108             if (cari == listMhs[mid].nim) {
109                 return (mid);
110             } else if (listMhs[mid].nim > cari) {
111                 return FindBinarySearch(cari, left, mid - 1);
112             } else {
113                 return FindBinarySearch(cari, mid + 1, right);
114             }
115         }
116         return -1;
117     }
118
119 }
```

2. Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas Mahasiswamain. Kemudian panggil method tampilposisi dan tampilData

```
89     System.out.println(x:"=====");
90     System.out.println(x:"Menggunakan binary Search");
91     posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1);
92     data.Tampilposisi(cari, posisi);
93     data.TampilData(cari, posisi);
94 }
95 }
```

6.3.2. Verifikasi Hasil Percobaan

```
-----  
Data Keseluruhan Mahasiswa:
```

```
Nim = 2017
```

```
Nama = Dewi Lestari
```

```
Umur = 23
```

```
IPK = 3.5
```

```
-----  
Nim = 2018
```

```
Nama = Sinta Sanjaya
```

```
Umur = 22
```

```
IPK = 4.0
```

```
-----  
Nim = 2019
```

```
Nama = Danang Adi
```

```
Umur = 22
```

```
IPK = 3.7
```

```
-----  
Nim = 2020
```

```
Nama = Budi Prakarsa
```

```
Umur = 20
```

```
IPK = 2.9
```

```
-----  
Nim = 2021
```

```
Nama = Vania Siti
```

```
Umur = 20
```

```
IPK = 3.0  
-----  
-----
```

```
-----  
Pencarian Data:
```

```
Masukkan Nim Mahasiswa yang dicari:
```

```
NIM :
```

```
2018
```

```
Menggunakan sequential search
```

```
data : 2018 ditemukan pada indeks1
```

```
Nim      : 2018
```

```
Nama     : Sinta Sanjaya
```

```
Umur     : 22
```

```
IPK      : 4.0
```

```
=====
```

```
Menggunakan binary Search
```

```
data : 2018 ditemukan pada indeks1
```

```
Nim      : 2018
```

```
Nama     : Sinta Sanjaya
```

```
Umur     : 22
```

```
IPK      : 4.0
```

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07> |
```

6.3.3. Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
4. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (missal : 20215, 20214, 20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard

Jawaban :

1. FindBinarySearch pada class PencarianMhs21. Berikut adalah potongan kode yang menunjukkan proses divide:

```
104     public int FindBinarySearch(int cari, int left, int right) {
105         int mid;
106         if (right >= left) {
107             mid = (left + right) / 2;
108             if (cari == listMhs[mid].nim) {
109                 return (mid);
110             } else if (listMhs[mid].nim > cari) {
111                 return FindBinarySearch(cari, left, mid - 1);
112             } else {
113                 return FindBinarySearch(cari, mid + 1, right);
114             }
115         }
116         return -1;
117     }
118
119 }
```

2. Proses conquer (penggabungan) dalam algoritma binary search dijalankan pada bagian berikut dari method FindBinarySearch pada class PencarianMhs21

```
108         if (cari == listMhs[mid].nim) {
109             return (mid);
110         }
```

3. Ya, program akan tetap berjalan meskipun data NIM yang dimasukkan tidak terurut. Namun, hasil yang dikeluarkan mungkin tidak akan selalu benar.
4. Jika data NIM yang dimasukkan sudah terurut dari terbesar ke terkecil, dan elemen yang dicari adalah 2020, hasil dari binary search akan tergantung pada implementasi algoritma pencarian. Dalam kasus ini, kita akan menggunakan implementasi binary search yang diberikan dalam kode program.
Namun, dalam implementasi yang diberikan dalam kode program, binary search tidak akan memberikan hasil yang benar jika data tidak terurut. Karena data yang dimasukkan sudah terurut dari terbesar ke terkecil, binary search kemungkinan besar akan memberikan hasil yang sesuai, yaitu menemukan bahwa elemen 2020 ada dalam array.

Berikut adalah hasil modifikasi berdasarkan code diatas

```
119 public int FindBinarySearch(int cari, int left, int right) {
120     while (left <= right) {
121         int mid = left + (right - left) / 2;
122
123         if (listMhs[mid].nim == cari) {
124             return mid; // Jika elemen ditemukan, kembalikan indeks
125         }
126
127         // Jika elemen di tengah lebih kecil dari nilai yang dicari, cari di setengah kiri
128         if (listMhs[mid].nim < cari) {
129             right = mid - 1;
130         }
131         // Jika elemen di tengah lebih besar dari nilai yang dicari, cari di setengah kanan
132         else {
133             left = mid + 1;
134         }
135     }
136     // Jika elemen tidak ditemukan, kembalikan -1
137     return -1;
138 }
139
140 }
```

5. Berikut adalah hasil modifikasi dari class main yang mana jumlah mahasiswa yang di input sesuai berdasarkan masukan keyboard

6.4. Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

Tambahkan class MergeSorting pada package tersebut

```
J MergeSorting21.java > MergeSorting21 > mergeSort(int[])
1 public class MergeSorting21 {
```

Pada class MergeSorting buatlah method mergeSort yang menerima parameter data array yang akan diurutkan

```
3 public void mergeSort(int[] data) {
4
5     Sort(data, left:0, data.length - 1);
6
7 }
```

Buatlah method merge untuk melakukan proses penggabungan data dari bagian kiri dan kanan.

```
9 public void Merge(int data[], int left, int middle, int right) {
10     int[] temp = new int[data.length];
11 }
```


Implementasikan proses merge sebagai berikut.

```
9      public void Merge(int data[], int left, int middle, int right) {
10          int[] temp = new int[data.length];
11
12          for (int i = left; i <= right; i++) {
13              temp[i] = data[i];
14          }
15
16          int a = left;
17          int b = middle + 1;
18          int c = left;
19
20          while (a <= middle && b <= right) {
21              if (temp[a] <= temp[b]) {
22                  data[c] = temp[a];
23                  a++;
24              } else {
25                  data[c] = temp[b];
26                  b++;
27              }
28
29              c++;
30          }
31
32          int s = middle - a;
33
34          for (int i = 0; i <= s; i++) {
35              data[c + i] = temp[a + i];
36          }
37      }
38
39  }
```

Buatlah method sort

```
41  ✓      public void Sort(int data[], int left, int right) {
42
```

Implementasikan kode berikut pada method sort

```
41  ✓      public void Sort(int data[], int left, int right) {
42
43  ✓          if (left < right) {
44              int middle = (left + right) / 2;
45              Sort(data, left, middle);
46              Sort(data, middle + 1, right);
47              Merge(data, left, middle, right);
48          }
49
50      }
51
```

Pada method mergeSort, panggil method sort dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir.

```
3     public void mergeSort(int[] data) {  
4  
5         Sort(data, left:0, data.length - 1);  
6  
7     }  
8
```

Tambahkan method printArray

```
52     public void PrintArray(int arr[]) {  
53         int n = arr.length;  
54  
55         for (int i = 0; i < n; i++) {  
56             System.out.print(arr[i] + " ");  
57         }  
58  
59         System.out.println();  
60     }  
61  
62 }
```

Sebagai langkah terakhir, deklarasikan data yang akan diurutkan kemudian panggil proses sorting pada class SortMain

```
J SortMain21.java > SortMain21  
1     public class SortMain21 {  
2  
3         Run | Debug  
4         public static void main(String[] args) {  
5             int data[] = { 10, 40, 30, 50, 70, 20, 100, 90 };  
6  
7             System.out.println(x:"=====");  
8             System.out.println(x:"          Sorting dengan merge sort");  
9             System.out.println(x:"=====");  
10            MergeSorting21 mSort = new MergeSorting21();  
11  
12            System.out.println(x:"          DATA AWAL");  
13            System.out.println(x:"=====");  
14            mSort.PrintArray(data);  
15            mSort.mergeSort(data);  
16  
17            System.out.println(x:"=====");  
18            System.out.println(x:"          Data setelah diurutkan");  
19            System.out.println(x:"=====");  
20            mSort.PrintArray(data);  
21            System.out.println(x:"=====");  
22        }  
23    }
```

6.4.2. Verifikasi Hasil Percobaan

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07> & 'C:\wCodeDetailsInExceptionMessages' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\workspaceStorage\87\bin' 'SortMain21'

=====
      Sorting dengan merge sort
=====
      DATA AWAL
=====
10 40 30 50 70 20 100 90
=====
      Data setelah diurutkan
=====
10 20 30 40 50 70 90 100
=====
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07>
```

6.5. Latihan Praktikum

1. Modifikasi percobaan searching diatas yang menggunakan Searching array of object dengan ketentuan berikut ini
 - Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma binary Search)
 - Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan

- MahasiswaMain_21.java

```

J MahasiswaMain_21.java > MahasiswaMain_21 > main(String[])
1  import java.util.Scanner;
2
3  public class MahasiswaMain_21 {
4      Run | Debug
      public static void main(String[] args) {
5
6          Scanner input18 = new Scanner(System.in);
7          Scanner input18Line = new Scanner(System.in);
8
9
10         System.out.println(x:"=====");
11         System.out.print(s:"Masukkan jumlah mahasiswa   : ");
12         int jumlahMahasiswa = input18.nextInt();
13         PencarianMahasiswa_21 data = new PencarianMahasiswa_21(jumlahMahasiswa);
14         System.out.println(x:"=====");
15
16         System.out.println(x:"=====");
17         System.out.println(x:"   Masukkan data mahasiswa secara urut dari NIM terkecil");
18         System.out.println(x:"=====");
19
20         for (int i = 0; i < jumlahMahasiswa; i++) {
21
22             System.out.print(s:"NIM\t : ");
23             int nim = input18.nextInt();
24             System.out.print(s:"Nama\t : ");
25             String nama = input18Line.nextLine();
26             System.out.print(s:"Umur\t : ");
27             int umur = input18.nextInt();
28             System.out.print(s:"IPK\t : ");
29             double ipk = input18.nextDouble();
30             System.out.println(x:"=====");
31
32             System.out.println(x:"=====");
33
34             Mahasiswa21 m = new Mahasiswa21(nim, nama, umur, ipk);
35             data.Tambah(m);
36
37             System.out.println(x:"=====");
38             System.out.println(x:"                           Data keseluruhan mahasiswa");
39             System.out.println(x:"=====");
40             data.Tampil();
41
42             System.out.println(x:"=====");
43             System.out.println(x:"                           Pencarian data mahasiswa");
44             System.out.println(x:"=====");
45             System.out.print(s:"Masukkan NIM mahasiswa yang ingin dicari : ");
46             int cari = input18.nextInt();
47
48             System.out.println(x:"=====");
49             System.out.println(x:"                           Menggunakan Sequential Search");
50             System.out.println(x:"=====");
51             int posisi = data.FindSequentialSearch(cari);
52
53             data.TampilPosisi(cari, posisi);
54             data.TampilData(cari, posisi);
55
56             data.TampilPosisi(cari, posisi);
57             data.TampilData(cari, posisi);
58
59             System.out.println(x:"=====");
60             System.out.println(x:"                           Menggunakan Binary Search");
61             System.out.println(x:"=====");
62             posisi = data.FindBinarySearch(cari, left:0, jumlahMahasiswa - 1);
63
64             data.TampilPosisi(cari, posisi);
65             data.TampilData(cari, posisi);
66         }
67     }

```


- Mahasiswa21.java

```
28 public class Mahasiswa21 {  
29  
30     int nim;  
31     String nama;  
32     int umur;  
33     double ipk;  
34  
35     Mahasiswa21(int nim, String nama, int umur, double ipk) {  
36  
37         this.nim = nim;  
38         this.nama = nama;  
39         this.umur = umur;  
40         this.ipk = ipk;  
41  
42     }  
43  
44     void Tampil() {  
45  
46         System.out.println("NIM      : " + nim);  
47         System.out.println("Nama    : " + nama);  
48         System.out.println("Umur   : " + umur);  
49         System.out.println("IPK    : " + ipk);  
50  
51     }  
52  
53 }
```

- PencarianMahasiswa_21.java

```
J PencarianMahasiswa_21.java > PencarianMahasiswa_21
1 public class PencarianMahasiswa_21 {
2     Mahasiswa21 listMahasiswa[];
3     int index;
4
5     public PencarianMahasiswa_21(int jumlahMahasiswa) {
6         listMahasiswa = new Mahasiswa21[jumlahMahasiswa];
7         index = 0;
8     }
9
10    void Tambah(Mahasiswa21 m) {
11        if (index < listMahasiswa.length) {
12            listMahasiswa[index] = m;
13            index++;
14        } else {
15            System.out.println(x:"Data sudah penuh !!!");
16        }
17    }
18
19
20    void Tampil() {
21        for (Mahasiswa21 m : listMahasiswa) {
22            m.Tampil();
23            System.out.println(x:"=====");
24        }
25    }
26
27    public int FindSequentialSearch(int cari) {
28        int posisi = -1;
29        for (int j = 0; j < listMahasiswa.length; j++) {
30            if (listMahasiswa[j].nim == cari) {
31                posisi = j;
32                break;
33            }
34        }
35        return posisi;
36    }
37
38    public int FindBinarySearch(int cari, int left, int right) {
39        int mid;
40        if (right >= left) {
41            mid = (left + right) / 2;
42            if (cari == listMahasiswa[mid].nim) {
43                return mid;
44            } else if (listMahasiswa[mid].nim < cari) { // Modifikasi pada tanda lebih dari (>) ke (<)
45                return FindBinarySearch(cari, left, mid - 1);
46            } else {
47                return FindBinarySearch(cari, mid + 1, right);
48            }
49        }
50        return -1;
51    }
52
53    public void TampilPosisi(int x, int pos) {
54        if (pos != -1) {
55
56            public void TampilPosisi(int x, int pos) {
57                if (pos != -1) {
58                    System.out.println("Data " + x + " ditemukan pada indeks " + pos);
59                } else {
60                    System.out.println("Data " + x + " tidak ditemukan");
61                }
62            }
63
64            public void TampilData(int x, int pos) {
65                if (pos != -1) {
66                    System.out.println("NIM\t : " + x);
67                    System.out.println("Nama\t : " + listMahasiswa[pos].nama);
68                    System.out.println("Umur\t : " + listMahasiswa[pos].umur);
69                    System.out.println("IPK\t : " + listMahasiswa[pos].ipk);
70                } else {
71                    System.out.println("Data " + x + " tidak ditemukan");
72                }
73            }
74        }
75    }
76}
```

- OutPut

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07> & 'C:\Program Files\Java\bin\java.exe' -cp 'C:\Users\HP\AppData\Roaming\Code\User\workspaceStorage\724fe32959bc91cd8f73' 'MahasiswaMain_21'
```

```
=====
Masukkan jumlah mahasiswa : 4
=====
Masukkan data mahasiswa secaraurut dari NIM terkecil
=====
NIM      : 12345
Nama     : robert
Umur     : 22
IPK      : 3,8
=====
NIM      : 12346
Nama     : jonh
Umur     : 23
IPK      : 3,7
=====
NIM      : 12347
Nama     : alex
Umur     : 21
IPK      : 4,0
=====
NIM      : 12348
Nama     : roya
Umur     : 21
IPK      : 4,0
=====
```

```
=====
Data keseluruhan mahasiswa
=====
NIM      : 12345
Nama     : robert
Umur     : 22
IPK      : 3.8
=====
NIM      : 12346
Nama     : jonh
Umur     : 23
IPK      : 3.7
=====
NIM      : 12347
Nama     : alex
Umur     : 21
IPK      : 4.0
=====
NIM      : 12348
Nama     : roya
Umur     : 21
IPK      : 4.0
=====
```

```
=====
Pencarian data mahasiswa
=====
Masukkan NIM mahasiswa yang ingin dicari : 12345
=====
Menggunakan Sequential Search
=====
Data 12345 ditemukan pada indeks 0
NIM      : 12345
Nama     : robert
Umur     : 22
IPK      : 3.8
=====
Menggunakan Binary Search
=====
Data 12345 tidak ditemukan
Data 12345 tidak ditemukan
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\Jobsheet 07> |
```