

**LAPORAN PRAKTIKUM JOBSHEET 6**  
**SEARCHING**  
**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**



**Disusun Oleh :**  
**Jami'atul Afifah (2341760102)**  
**SIB-1F**

**PROGRAM STUDI D4 SISTEM INFOEMASI BISNIS**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## Sequential Search Menggunakan Array

1. Buat folder baru dengan nama Praktikum06. Buat file dengan nama Sorting.java
2. Tambahkan method sequentialSearch() yang melakukan pencarian data bertipe integer di dalam

```
public class Sorting {  
    public static void sequentialSearch(int[] arr, int key) {  
        for (int i = 0; i < arr.length; i++) {  
            if (arr[i] == key) {  
                System.out.println("Data ditemukan pada indeks ke-" + i);  
                return; // Menghentikan pencarian setelah data ditemukan  
            }  
        }  
        System.out.println(x:"Data tidak ditemukan");  
    }  
}
```

3. Tambahkan fungsi main sebagai berikut

```
Run | Debug  
public static void main(String[] args) {  
    int[] daftarNilai = { 10, 5, 20, 15, 80, 45 };  
    sequentialSearch(daftarNilai, key:5);  
}
```

4. Compile dan run program

```
a16e2537ab37f0870dbef6f85aa975\rednat.  
Data ditemukan pada indeks ke-1  
PS D:\Matkul\SEM 2\ASD\jobsheet7> □
```

## Sequential Search Menggunakan Array of Object

### Langkah-langkah Percobaan Sequential Search

1. Buatlah Project baru pada vscode dengan nama TestSearching
2. Kemudian buat packages baru dengan nama minggu7.
3. Buat class Mahasiswa, kemudian deklarasikan atribut berikut ini

```
public class Mahasiswa {  
    int nim;  
    String nama;  
    int umur;  
    double ipk;
```

4. Buatlah konstruktor dengan nama Mahasiswa dengan parameter (int ni, String n, int u, double i) kemudian isi konstruktor tersebut dengan kode berikut!

```
public Mahasiswa(int ni, String n, int u, double i) {  
    nim = ni;  
    nama = n;  
    umur = u;  
    ipk = i;  
}
```

5. Buatlah method tampil bertipe void

```
public void tampil() {  
    System.out.println("Nim = " + nim);  
    System.out.println("Nama = " + nama);  
    System.out.println("Umur = " + umur);  
    System.out.println("IPK = " + ipk);  
}
```

6. Buat class baru dengan nama PencarianMhs seperti di bawah ini!

```
public class PencarianMhs {  
    Mahasiswa listMhs[] = new Mahasiswa[5];  
    int idx;
```

7. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.\

```
public void tambah(Mahasiswa m) {  
    if (idx < listMhs.length) {  
        listMhs[idx] = m;  
        idx++;  
    } else {  
        System.out.println(x:"Data sudah penuh !!");  
    }  
}
```

8. Tambahkan method tampil() di dalam class PencarianMhs! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.

```
public void tampil() {  
    for (Mahasiswa m : listMhs) {  
        m.tampil();  
        System.out.println(x:"-----");  
    }  
}
```

9. Tambahkan method FindSeqSearch bertipe integer dengan parameter cari bertipe integer. Kemudian Deklarasikan isi method FindSeqSearch dengan algoritma pencarian data menggunakan teknik sequential searching.

```
public int FindSeqSearch(int cari) {  
    int posisi = -1;  
    for (int j = 0; j < listMhs.length; j++) {  
        if (listMhs[j].nim == cari) {  
            posisi = j;  
            break;  
        }  
    }  
    return posisi;  
}
```

10. Buatlah method `Tampilpoisisi` bertipe `void` dan Deklarasikan isi dari method `Tampilpoisisi`.

```
public void Tampilpoisisi(int x, int pos) {
    if (pos != -1) {
        System.out.println("data: " + x + " ditemukan pada indeks " + pos);
    } else {
        System.out.println("data " + x + " tidak ditemukan");
    }
}
```

11. Buatlah method `TampilData` bertipe `void` dan Deklarasikan isi dari method `TampilData`

```
public void TampilData(int x, int pos) {
    if (pos != -1) {
        System.out.println("Nim\t: " + x);
        System.out.println("Nama\t: " + listMhs[pos].nama);
        System.out.println("Umur\t: " + listMhs[pos].umur);
        System.out.println("IPK\t: " + listMhs[pos].ipk);
    } else {
        System.out.println("data " + x + " tidak ditemukan");
    }
}
```

12. Buatlah class baru dengan nama `MahasiswaMain` tambahkan method `main` seperti pada gambar berikut!

```
public class MahasiswaMain {
    Run | Debug
    public static void main(String[] args) {
```

13. Di dalam method `main()`, buatlah sebuah objek `PencarianMhs` dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi `tambah` pada objek `PencarianMhs`.

```
Scanner s = new Scanner(System.in);
Scanner sl = new Scanner(System.in);

PencarianMhs data = new PencarianMhs();
int jumMhs = 5;

System.out.println(x: "-----");
System.out.println(x: "Masukkan data mahasiswa secara Urut dari Nim Terkecil");

for (int i = 0; i < jumMhs; i++) {
    System.out.println(x: "-----");
    System.out.print(s: "Nim\t: ");
    int nim = s.nextInt();
    System.out.print(s: "Nama\t: ");
    String nama = sl.nextLine();
    System.out.print(s: "Umur\t: ");
    int umur = s.nextInt();
    System.out.print(s: "IPK\t: ");
    double ipk = s.nextDouble();
    Mahasiswa m = new Mahasiswa(nim, nama, umur, ipk);
    data.tambah(m);
}
```

14. Panggil method tampil() untuk melihat semua data yang telah dimasukan.\

```
System.out.println(x:"-----");
System.out.println(x:"Data keseluruhan Mahasiswa : ");
data.tampil();
```

15. Untuk melakukan pencarian berdasarkan NIM mahasiswa. Buatlah variable cari yang dapat menampung masukan dari keyboard lalu panggil method FindSeqSearch dengan isi parameternya adalah variable cari.

```
System.out.println(x:"-----");
System.out.println(x:"-----");
System.out.println(x:"Pencarian Data: ");
System.out.println(x:"Masukkan Nim Mahasiswa yang dicari: ");
System.out.print(s:"NIM: ");
int cari = s.nextInt();
System.out.println(x:"menggunakan sequential Search");
int posisi = data.FindSeqSearch(cari);
```

16. Lakukan pemanggilan method Tampilposisi dari class PencarianMhs.

17. Lakukan pemanggilan method TampilData dari class PencarianMhs.

```
data.Tampilposisi(cari, posisi);
data.TampilData(cari, posisi);
```

### Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini.

```
-----
Nim = 2018
Nama = sinta
Umur = 22
IPK = 4.0
-----
Nim = 2019
Nama = danang
Umur = 24
IPK = 2.9
-----
Nim = 2020
Nama = budi
Umur = 20
IPK = 3.7
-----
Nim = 2021
Nama = vania
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
Umur     : 22
IPK      : 4.0
PS D:\Matkul\SEM 2\ASD\jobsheet7> 
```

## Pertanyaan

- 1) Lakukan perubahan array daftarNilai pada fungsi main().

```
15     public static void main(String[] args) {
16         int[] daftarNilai = { 10, 5, 20, 15, 5, 45 };
17         sequentialSearch(daftarNilai, key:5);
18     }
19 }
20
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS D:\Matkul\SEM 2\ASD\jobsheet7> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' 'C:\Users\Afifah\AppData\Roaming\Code\User\workspaceStorage\9da16e2537ab37f08870dbef6f85aa975'

Data ditemukan pada indeks ke-1

- 2) Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs
- TampilData(int x, int pos): Method ini bertugas untuk menampilkan data mahasiswa yang ditemukan berdasarkan NIM. Jika data dengan NIM yang dicari ditemukan, method ini akan menampilkan semua atribut mahasiswa (NIM, nama, umur, dan IPK). Jika data tidak ditemukan, method ini akan menampilkan pesan bahwa data tidak ditemukan.
  - Tampilposisi(int x, int pos): Method ini bertugas untuk menampilkan posisi indeks dari data mahasiswa yang ditemukan berdasarkan NIM. Jika data dengan NIM yang dicari ditemukan, method ini akan menampilkan pesan bahwa data ditemukan beserta indeksnya. Jika data tidak ditemukan, method ini akan menampilkan pesan bahwa data tidak ditemukan.
- 3) Jelaskan fungsi break pada kode program dibawah ini!

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

Fungsi break pada kode program tersebut digunakan untuk menghentikan iterasi pada loop for saat kondisi pencarian data terpenuhi. Setelah data ditemukan, break digunakan untuk keluar dari loop for, sehingga tidak perlu lagi melakukan iterasi pada elemen array yang tersisa.

- 4) Jika Data Nim yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Mengapa demikian!
- Jika data NIM yang dimasukkan tidak terurut dari kecil ke besar, program masih dapat berjalan namun hasilnya mungkin tidak benar. Algoritma pencarian Sequential Search tidak memerlukan data yang terurut, tetapi ketika data tidak terurut, waktu yang diperlukan untuk menemukan data tertentu dapat lebih lama karena harus memeriksa setiap elemen array.

## Searching / Pencarian Menggunakan Binary Search

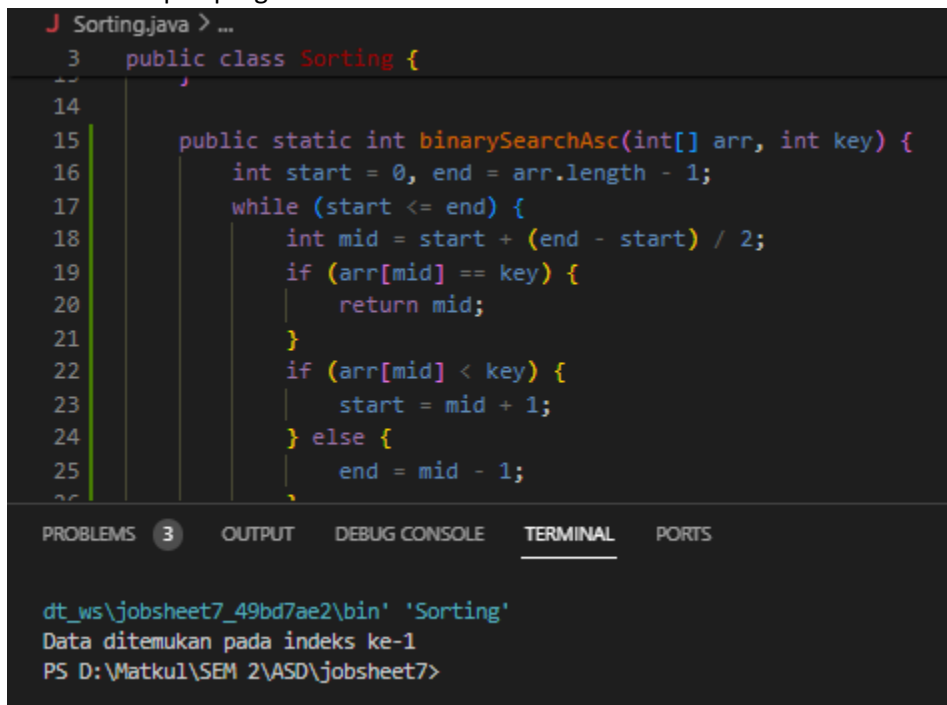
1. Tambahkan method `binarySearchAsc()` pada file `Sorting.java`

```
public static int binarySearchAsc(int[] arr, int key) {
    int start = 0, end = arr.length - 1;
    while (start <= end) {
        int mid = start + (end - start) / 2;
        if (arr[mid] == key) {
            return mid;
        }
        if (arr[mid] < key) {
            start = mid + 1;
        } else {
            end = mid - 1;
        }
    }
    return -1;
}
```

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

```
public static void main(String[] args) {
    int[] sortedNilai = { 5, 5, 10, 20, 30, 40, 50};
    int index = binarySearchAsc(sortedNilai, key:5);
    if (index != -1) {
        System.out.println("Data ditemukan pada indeks ke-" + index);
    } else {
        System.out.println(x:"Data tidak ditemukan");
    }
}
```

3. Run dan compile program



The screenshot shows an IDE window titled "Sorting.java > ...". The code editor displays the `Sorting` class with the `binarySearchAsc` method and the `main` method. The `main` method creates an array `sortedNilai` with values `{ 5, 5, 10, 20, 30, 40, 50}` and calls `binarySearchAsc(sortedNilai, key:5)`. The IDE interface includes tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". The "TERMINAL" tab is active, showing the output of the program: `dt_ws\jobsheet7_49bd7ae2\bin' 'Sorting'`, `Data ditemukan pada indeks ke-1`, and the command prompt `PS D:\Matkul\SEM 2\ASD\jobsheet7>`.

## 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.

```
public int FindBinarySearch(int cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (left + right) / 2;
        if (cari == listMhs[mid].nim) {
            return mid;
        } else if (listMhs[mid].nim > cari) {
            return FindBinarySearch(cari, left, mid - 1);
        } else {
            return FindBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}
```

2. Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas

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

3. Jalankan dan amati hasilnya.

## Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini.

```
Nim = 2017
Nama = dewi
Umur = 23
IPK = 3.5
-----
Nim = 2018
Nama = sinta
Umur = 22
IPK = 4.0
-----
Nim = 2019
Nama = danang
Umur = 22
IPK = 3.7
-----
Nim = 2020
Nama = budi
Umur = 20
IPK = 2.9
-----
Nim = 2021
Nama = vania
Umur = 23
IPK = 3.4
-----
Pencarian Data:
Masukkan Nim Mahasiswa yang dicari:
NIM: 2019
menggunakan sequential Search
data: 2019 ditemukan pada indeks 2
Nim      : 2019
Nama     : danang
Umur     : 22
IPK      : 3.7
-----
menggunakan binary Search
data: 2019 ditemukan pada indeks 2
Nim      : 2019
Nama     : danang
Umur     : 22
IPK      : 3.7
```



## Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!

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

Pada baris tersebut, variabel mid digunakan untuk membagi rentang pencarian menjadi dua bagian.

2. Tunjukkan pada kode program yang mana proses conquer dijalankan!

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

Pada baris tersebut, jika elemen yang dicari ditemukan pada indeks mid, maka proses conquer dilakukan dengan mengembalikan nilai mid.

3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!

Jika data Nim yang dimasukkan tidak urut, program masih dapat berjalan namun hasil pencarian menggunakan binary search mungkin tidak akurat. Hal ini disebabkan oleh asumsi binary search yang memerlukan data terurut. Jika data tidak terurut, binary search tidak dapat menemukan elemen yang dicari secara benar.

4. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (misal : 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 search agar hasilnya sesuai

```
public int FindBinarySearch(int cari, int left, int right) {  
    int mid;  
    if (right >= left) {  
        mid = (left + right) / 2;  
        if (cari == listMhs[mid].nim) {  
            return mid;  
        } else if (listMhs[mid].nim < cari) {  
            return FindBinarySearch(cari, left, mid - 1);  
        } else {  
            return FindBinarySearch(cari, mid + 1, right);  
        }  
    }  
    return -1;  
}
```

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

```
public class MahasiswaMain {  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
        Scanner sl = new Scanner(System.in);  
  
        PencarianMhs data = new PencarianMhs();  
  
        System.out.println(x:"Masukkan jumlah mahasiswa:");  
        int jumMhs = s.nextInt();  
  
        System.out.println(x:"-----");  
        System.out.println(x:"Masukkan data mahasiswa secara Urut dari Nim Terkecil");  
  
        for (int i = 0; i < jumMhs; i++) {  
            System.out.println(x:"-----");  
            System.out.print(s:"Nim\t: ");  
            int nim = s.nextInt();  
            System.out.print(s:"Nama\t: ");  
            String nama = sl.nextLine();  
            System.out.print(s:"Umur\t: ");  
            int umur = s.nextInt();  
            System.out.print(s:"IPK\t: ");  
            double ipk = s.nextDouble();  
            Mahasiswa m = new Mahasiswa(nim, nama, umur, ipk);  
            data.tambah(m);  
        }  
    }  
}
```

## Percobaan Pengayaan Divide and Conquer

### Langkah-langkah Percobaan Merge Sort

1. Buatlah Package baru pada NetBeans dengan nama MergeSortTest
2. Tambahkan class MergeSorting pada package tersebut
3. Pada class MergeSorting buatlah method mergeSort yang menerima parameter data array yang akan diurutkan

```
public void mergeSort(int[] data) {
```

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

```
private void sort(int data[], int left, int right)
```

5. Implementasikan proses merge sebagai berikut.

```
public void merge(int data[], int left, int middle, int right) {  
    int[] temp = new int[data.length];  
    for (int i = left; i <= right; i++) {  
        temp[i] = data[i];  
    }  
    int a = left;  
    int b = middle + 1;  
    int c = left;  
  
    // membandingkan setiap bagian  
    while (a <= middle && b <= right) {  
        if (temp[a] <= temp[b]) {  
            data[c] = temp[a];  
            a++;  
        } else {  
            data[c] = temp[b];  
            b++;  
        }  
        c++;  
    }  
  
    int s = middle - a;  
    for (int i = 0; i <= s; i++) {  
        data[c + i] = temp[a + i];  
    }  
}
```

6. Buatlah method sort
7. Implementasikan kode berikut pada method sort

```
// membagi menjadi 2 bagian dan dibagi kembali hingga tidak dapat dibagi kembali  
private void sort(int data[], int left, int right) {  
    if (left < right) {  
        int middle = (left + right) / 2;  
        sort(data, left, middle);  
        sort(data, middle + 1, right);  
        merge(data, left, middle, right);  
    }  
}
```

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

```
public void printArray(int arr[]) {  
    int n = arr.length;  
    for (int i = 0; i < n; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
}
```

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

```
public class SortMain {  
    Run | Debug  
    public static void main(String[] args) {  
        int data[] = {10, 40, 30, 50, 70, 20, 100, 90};  
        System.out.println(x:"sorting dengan merge sort");  
        MergeSortTest mSort= new MergeSortTest ();  
        System.out.println(x:"data awal");  
        mSort.printArray (data);  
        mSort.mergeSort (data) ;  
        System.out.println(x:"setelah diurutkan");  
        mSort.printArray (data) ;  
    }  
}
```

### Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini

```
sorting dengan merge sort  
data awal  
10 40 30 50 70 20 100 90  
setelah diurutkan  
10 20 30 40 50 70 90 100  
PS D:\Matkul\SEM 2\ASD\jobsheet7>
```

## Latihan Praktikum

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!

MainMaha14.java

```
J MainMaha14.java > ...
1  import java.util.Scanner;
2
3  public class MainMaha14 {
4      Run | Debug
5      public static void main(String[] args) {
6          Scanner s = new Scanner(System.in);
7          Scanner s1 = new Scanner(System.in);
8
9          pencariaha14 data = new pencariaha14();
10
11          System.out.println(x:"Masukkan jumlah mahasiswa:");
12          int jumMhs = s.nextInt();
13
14          System.out.println(x:"-----");
15          System.out.println(x:"Masukkan data mahasiswa secara Urut dari Nim Terkecil");
16
17          for (int i = 0; i < jumMhs; i++) {
18              System.out.println(x:"-----");
19              System.out.print(s:"Nim\t: ");
20              int nim = s.nextInt();
21              System.out.print(s:"Nama\t: ");
22              String nama = s1.nextLine();
23              System.out.print(s:"Umur\t: ");
24              int umur = s.nextInt();
25              System.out.print(s:"IPK\t: ");
26              double ipk = s.nextDouble();
27              maha14 m = new maha14(nim, nama, umur, ipk);
28              data.tambah(m);
29          }
30
31          System.out.println(x:"-----");
32          System.out.println(x:"Data keseluruhan Mahasiswa : ");
33          data.tampil();
34
35          System.out.println(x:"-----");
36          System.out.println(x:"Pencarian Data: ");
37          System.out.println(x:"Masukkan Nama Mahasiswa yang dicari: ");
38          System.out.print(s:"Nama: ");
39          String cari = s.next();
40
41          System.out.println(x:"Mencari menggunakan binary Search");
42          data.searchByName(cari);
43      }
44  }
```

## Pencariaha14.java

```
J: pencariaha14.java > ...
1 public class pencariaha14 {
2     maha14 listMhs[] = new maha14[5];
3     int idx;
4
5     public void tambah(maha14 m) {
6         if (idx < listMhs.length) {
7             listMhs[idx] = m;
8             idx++;
9         } else {
10             System.out.println(x:"Data sudah penuh !!");
11         }
12     }
13
14     public void tampil() {
15         for (int i = 0; i < idx; i++) {
16             if (listMhs[i] != null) {
17                 listMhs[i].tampil();
18                 System.out.println(x:"-----");
19             }
20         }
21     }
22
23     public int FindBinarySearchByName(String cari, int left, int right, int count) {
24         if (right >= left) {
25             int mid = (left + right) / 2;
26             int comparison = listMhs[mid].nama.compareTo(cari);
27
28             if (comparison < 0) {
29                 return FindBinarySearchByName(cari, mid + 1, right, count);
30             } else if (comparison > 0) {
31                 return FindBinarySearchByName(cari, left, mid - 1, count);
32             } else {
33                 return mid;
34             }
35         }
36         return -1;
37     }
}
```

```
1 public class pencariaha14 {
38
39     public void searchByName(String cari) {
40         int pos = FindBinarySearchByName(cari, left:0, idx - 1, count:0);
41
42         if (pos != -1) {
43             // Check if there are multiple results
44             int count = 1;
45             for (int i = pos - 1; i >= 0 && listMhs[i].nama.equalsIgnoreCase(cari); i--) {
46                 count++;
47             }
48             for (int i = pos + 1; i < idx && listMhs[i].nama.equalsIgnoreCase(cari); i++) {
49                 count++;
50             }
51
52             if (count > 1) {
53                 System.out.println(x:"Data yang ditemukan lebih dari 1");
54             }
55
56             Tampilpoisisi(cari, pos);
57             TampilData(cari, pos);
58         } else {
59             System.out.println("Data " + cari + " tidak ditemukan");
60         }
61     }
62
63     public void Tampilpoisisi(String x, int pos) {
64         if (pos != -1) {
65             System.out.println("Data: " + x + " ditemukan pada indeks " + pos);
66         } else {
67             System.out.println("Data " + x + " tidak ditemukan");
68         }
69     }
70
71     public void TampilData(String x, int pos) {
72         if (pos != -1) {
73             System.out.println("Nim\t: " + listMhs[pos].nim);
74             System.out.println("Nama\t: " + x);
75             System.out.println("Umur\t: " + listMhs[pos].umur);
76             System.out.println("IPK\t: " + listMhs[pos].ipk);
77         } else {
78             System.out.println("Data " + x + " tidak ditemukan");
79         }
80     }
81 }
82
```

Maha14.java

```
J maha14.java > ...
1  public class maha14 {
2      int nim;
3      String nama;
4      int umur;
5      double ipk;
6
7      public maha14(int ni, String n, int u, double i) {
8          nim = ni;
9          nama = n;
10         umur = u;
11         ipk = i;
12     }
13
14     public void tampil() {
15         System.out.println("Nim = " + nim);
16         System.out.println("Nama = " + nama);
17         System.out.println("Umur = " + umur);
18         System.out.println("IPK = " + ipk);
19     }
20 }
21
```