

# **Laporan Teori Algoritma dan Struktur Data**

## **Jobsheet 7 – Searching**

**Dosen Pengampu : Ibu Triana Fatmawati, S.T., M.T**



**Nama : Annisa**  
**Nim : 2341760032**  
**Prodi : D-IV Sistem Informasi Bisnis**  
**Absen : 04**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

## 6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

### 6.2.1 Sequential Search Menggunakan Array

```
1  /**
2   * Sorting04
3   */
4  public class Sorting04 {
5
6      public static void sequentialSearch(int[] arr, int key) {
7          for (int i = 0; i < arr.length; i++) {
8              if (arr[i] == key) {
9                  System.out.println("Data ditemukan pada indeks ke-" + i);
10                 return;
11             }
12         }
13         System.out.println("Data tidak ditemukan");
14     }
15
16     public static void main(String[] args) {
17         int[] daftarNilai = {10, 5, 20, 15, 80, 45};
18         sequentialSearch(daftarNilai, 5);
19     }
20 }
```

Data ditemukan pada indeks ke-1

C:\Users\user\Documents\ASD Jobsheet 7>

### 6.2.2 Sequential Search Menggunakan Array of Object

- Langkah-langkah Percobaan Sequential Search

```
1  /**
2   * Mahasiswa04
3   */
4  public class Mahasiswa04 {
5
6      String nama;
7      int nim, umur;
8      double ipk;
9
10     Mahasiswa04 (String n, int ni, int u, double i) {
11         nama = n;
12         nim = ni;
13         umur = u;
14         ipk = i;
15     }
16     void tampil() {
17         System.out.println("Nim = " + nim);
18         System.out.println("Nama = " + nama);
19         System.out.println("Umur = " + umur );
20         System.out.println("IPK = " + ipk);
21     }
22 }
```

```

1  /**
2   * PencarianMhs04
3   */
4  public class PencarianMhs04 {
5
6      Mahasiswa04 listMhs[] = new Mahasiswa04[5];
7      int idx;
8
9      void tambah(Mahasiswa04 m) {
10         if(idx < listMhs.length) {
11             listMhs[idx] = m;
12             idx++;
13         } else {
14             System.out.println("Data sudah penuh!!");
15         }
16     }
17     void tampil() {
18         for(Mahasiswa04 m : listMhs) {
19             m.tampil();
20             System.out.println("=====");
21         }
22     }
23
24     public int FindSeqSearch (int cari) {
25         int posisi = -1;
26         for (int j = 0; j < listMhs.length; j++) {
27             if (listMhs[j].nim == cari) {
28                 posisi = j;
29                 break;
30             }
31         }
32         return posisi;
33     }
34     public void TampilPosisi (int x, int pos) {
35         if (pos!= -1 ) {
36             System.out.println("Data : " + x + "ditemukan pada indeks " + pos);
37         } else {
38             System.out.println("Data " + x + "tidak ditemukan");
39         }
40     }
41     public void TampilData (int x, int pos) {
42         if (pos!= -1) {
43             System.out.println("Nim\t : " + x);
44             System.out.println("Nama\t : " + listMhs[pos].nama);
45             System.out.println("Umur\t : " + listMhs[pos].umur);
46             System.out.println("IPK\t : " + listMhs[pos].ipk);
47         } else {
48             System.out.println("data " + x + "tidak ditemukan ");
49         }
50     }
51 }

```

```

1  import java.util.Scanner;
2  /**
3   * MahasiswaMain04
4   */
5  public class MahasiswaMain04 {
6
7      public static void main(String[] args) {
8          Scanner s = new Scanner(System.in);
9          Scanner sl = new Scanner(System.in);
10
11          PencarianMhs04 data = new PencarianMhs04();
12          int jumMhs = 5;
13
14          System.out.println("=====");
15          System.out.println("Masukkan data mahasiswa secara urut dari NIM terkecil : ");
16          for (int i = 0; i < jumMhs; i++) {
17              System.out.println("=====");
18              System.out.print("Nim\t : ");
19              int nim = s.nextInt();
20              System.out.print("Nama\t : ");
21              String nama = sl.nextLine();
22              System.out.print("Umur\t : ");
23              int umur = s.nextInt();
24              System.out.print("IPK\t : ");
25              double ipk = s.nextDouble();
26
27              Mahasiswa04 m = new Mahasiswa04(nama, nim, umur, ipk);
28              data.tambah(m);
29
30              System.out.println("=====");
31              System.out.println("Data Keseluruhan Mahasiswa : ");
32              data.tampil();
33
34              System.out.println("=====");
35              System.out.println("=====");
36              System.out.println("Pencarian data : ");
37              System.out.println("Masukkan NIM Mahasiswa yang dicari: ");
38              System.out.print("NIM : ");
39              int cari = s.nextInt();
40              System.out.println("Menggunakan sequential Search");
41              int posisi = data.FindSeqSearch(cari);
42              data.TampilPosisi(cari, posisi);
43              data.TampilData(cari, posisi);
44
45          }
46      }
47  }

```

## 6.2.2. Verifikasi Hasil Percobaan

```

=====
Masukkan data mahasiswa secara urut dari NIM terkecil :
=====

```

```

Nim      : 2017
Nama     : Dewi Lestari
Umur     : 23
IPK      : 3.5
=====

```

```

Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4
=====

```

```

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
=====

```

```

=====
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 indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0

C:\Users\user\Documents\ASD Jobsheet 7>

```

### 6.2.3. 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);
}

```

```

1 Mahasiswa04 listMhs[] = new Mahasiswa04[15];
2 int idx;

```

#### 2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs

- Metode Tampilposisi bertanggung jawab untuk menampilkan hanya posisi (indeks) dari data yang ditemukan dalam array.  
Jika posisi yang diberikan bukan -1 (menunjukkan bahwa data ditemukan), maka metode ini mencetak pesan yang menyatakan bahwa data ditemukan pada indeks tertentu.  
Jika posisi adalah -1 (menunjukkan bahwa data tidak ditemukan), maka metode ini mencetak pesan yang menyatakan bahwa data tidak ditemukan.
- Metode TampilData bertanggung jawab untuk menampilkan seluruh data yang terkait dengan posisi yang diberikan.

Jika posisi yang diberikan bukan -1 (menunjukkan bahwa data ditemukan), maka metode ini mencetak informasi lengkap tentang data tersebut, seperti Nim, Nama, Umur, dan IPK dari elemen array yang sesuai dengan posisi.

Jika posisi adalah -1 (menunjukkan bahwa data tidak ditemukan), maka metode ini mencetak pesan yang menyatakan bahwa data tidak ditemukan.

### **3. Jelaskan fungsi break pada kode program dibawah ini!**

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

Fungsi dari break adalah untuk menghentikan eksekusi dari loop (baik itu for, while, atau do-while) secara paksa ketika suatu kondisi tertentu terpenuhi, tanpa harus menunggu loop mencapai kondisi terminasi normalnya.

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

program masih dapat berjalan, tetapi efisiensi pencarian dapat terpengaruh jika data tidak terurut. Untuk mencari hasil yang benar dalam pencarian sequential search, data biasanya harus terurut.

## 6.3. Searching / Pencarian Menggunakan Binary Search

### 6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

```
1  /**
2   * Sorting04
3   */
4  public class Sorting04 {
5
6      public static void sequentialSearch(int[] arr, int key) {
7          for (int i = 0; i < arr.length; i++) {
8              if (arr[i] == key) {
9                  System.out.println("Data ditemukan pada indeks ke-" + i);
10                 return;
11             }
12         }
13         System.out.println("Data tidak ditemukan");
14     }
15
16     public static int binnarySearchAsc(int[] arr, int key) {
17         int start = 0, end = arr.length - 1;
18
19         while (start <= end) {
20             int mid = start + (end - start) / 2;
21
22             if (arr[mid] == key) {
23                 return mid;
24             }
25             if (arr[mid] < key) {
26                 start = mid + 1;
27             } else {
28                 end = mid + 1;
29             }
30         }
31         return -1;
32     }
33
34
35     public static void main(String[] args) {
36         int[] daftarNilai = { 10, 5, 20, 15, 80, 45 };
37         sequentialSearch(daftarNilai, 5);
38
39         int[] sortedNilai = { 5, 5, 10, 20, 30, 40, 50 };
40         int index = binnarySearchAsc(sortedNilai, 5);
41
42         if (index != -1) {
43             System.out.println("Data ditemukan pada indeks ke-" + index);
44         } else {
45             System.out.println("Data tidak ditemukan ");
46         }
47     }
48 }
49
50
```

```
Data ditemukan pada indeks ke-1
Data ditemukan pada indeks ke-1
```

```
C:\Users\user\Documents\ASD Jobsheet 7>
```

### 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

```
1 public int FindBinarySearch (int cari, int left, int right) {
2     int mid;
3     if (right >= left ) {
4         mid = (left + right) / 2;
5         if (cari == listMhs[mid].nim) {
6             return (mid);
7         } else if (listMhs[mid].nim > cari) {
8             return FindBinarySearch(cari, left, mid - 1 );
9         } else {
10            return FindBinarySearch(cari, mid + 1, right);
11        }
12    }
13    return -1;
14 }
15 }
```

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

```
1 System.out.println("=====");
2     System.out.println("Menggunakan binary Search");
3     posisi = data.FindBinarySearch(cari, 0, jumMhs-1);
4     data.TampilPosisi(cari, posisi);
5     data.TampilData(cari, posisi);
6
7 }
```

3. Jalankan dan amati hasilnya.

#### 6.3.2. Verifikasi Hasil Percobaan

Masukkan data mahasiswa secara urut dari NIM terkecil :

```
=====
Nim      : 2017
Nama     : Dewi Lestari
Umur     : 23
IPK      : 3.5
=====
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4
=====
Nim      : 2019
Nama     : Donang 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
=====
```



```
=====
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 Aji
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 indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
=====
Menggunakan binary Search
Data : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
=====
C:\Users\user\Documents\ASD Jobsheet 7>
```

### 6.3.3. Pertanyaan

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

```
1  if (right >= left ) {
2      mid = (left + right) / 2;
3      if (cari == listMhs[mid].nim) {
4          return (mid);
```

Proses divide: Mencari nilai tengah untuk membagi array menjadi dua bagian yaitu pada  $mid = (left + right) / 2$ ;

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

A screenshot of a code editor with a dark background and light-colored text. The code is a recursive function `FindBinarySearch`. It shows the 'conquer' step of a binary search algorithm, where the search range is updated based on whether the current element is greater or less than the target. The code is as follows:

```
1 } else if (listMhs[mid].nim > cari) {
2     return FindBinarySearch(cari, left, mid - 1);
3 } else {
4     return FindBinarySearch(cari, mid + 1, right);
5 }
```

Proses conquer: Pemanggilan rekursif untuk mencari di bagian kiri array terdapat pada (cari , left, mid-1) Proses conquer: Pemanggilan rekursif untuk mencari di bagian kanan array terdapat pada (cari , mid + 1, right)

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

Tidak dapat berjalan, karena metode pencarian Binary Search mensyaratkan bahwa data harus diurutkan terlebih dahulu sebelum pencarian dilakukan. Jika data tidak diurutkan, pencarian biner tidak akan menghasilkan hasil yang benar. Ketika mencoba melakukan pencarian biner pada data yang tidak terurut, program mungkin akan menghasilkan output yang tidak konsisten atau bahkan salah.

## 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 search agar hasilnya sesuai

A screenshot of a code editor with a dark background and light-colored text. The code is a recursive function `FindBinarySearch` in Java. It shows the full implementation of a binary search algorithm, including the base case and the recursive calls. The code is as follows:

```
1 public int FindBinarySearch (int cari, int left, int right) {
2     int mid;
3     if (right >= left ) {
4         mid = (left + right) / 2;
5         if (cari == listMhs[mid].nim) {
6             return (mid);
7         } else if (listMhs[mid].nim < cari) {
8             return FindBinarySearch(cari, left, mid - 1);
9         } else {
10            return FindBinarySearch(cari, mid + 1, right);
11        }
12    }
13    return -1;
14 }
```

kode yang diubah terdapat pada pada else if (listMhs[mid].nim < cari) agar dapat mencari NIM dari yg terbesar ke yang terkecil.

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

Berikut merupakan hasil modifikasinya

```
1  /**
2   * PencarianMhs04
3   */
4  public class PencarianMhs04 {
5      Mahasiswa04 listMhs[];
6      int idx;
7
8      public PencarianMhs04(int maxSize) {
9          listMhs = new Mahasiswa04[maxSize];
10     }
11     void tambah(Mahasiswa04 m) {
12         if(idx < listMhs.length) {
13             listMhs[idx] = m;
14             idx++;
15         } else {
16             System.out.println("Data sudah penuh!!");
17         }
18     }
19
20     void tampil() {
21         for (int i = 0; i < idx; i++) {
22             listMhs[i].tampil();
23             System.out.println("-----");
24         }
25     }
26
27     public int FindSeqSearch (int cari) {
28         int posisi = -1;
29         for (int j = 0; j < idx; j++) {
30             if (listMhs[j].nim == cari) {
31                 posisi = j;
32                 break;
33             }
34         }
35         return posisi;
36     }
37     public int FindBinarySearch(int cari, int left, int right) {
38         int mid;
39         if (right >= left) {
40             mid = (left + right) / 2;
41             if (cari == listMhs[mid].nim) {
42                 return (mid);
43             } else if (listMhs[mid].nim < cari) {
44                 return FindBinarySearch(cari, left, mid - 1);
45             } else {
46                 return FindBinarySearch(cari, mid + 1, right);
47             }
48         }
49         return -1;
50     }
51     public void TampilPosisi (int x, int pos) {
52         if (pos != -1) {
53             System.out.println("data : " + x + "\nditemukan pada indeks " + (
54                 pos+1));
55         } else {
56             System.out.println("data " + x + "tidak ditemukan");
57         }
58     }
59
60     public void TampilData (int x, int pos) {
61         if (pos != -1) {
62             System.out.println("Nim\t : " + x);
63             System.out.println("Nama\t : " + listMhs[pos].nama);
64             System.out.println("Umur\t : " + listMhs[pos].umur);
65             System.out.println("IPK\t : " + listMhs[pos].ipk);
66         } else {
67             System.out.println("data " + x + "tidak ditemukan");
68         }
69     }
70 }
```

```

1  import java.util.Scanner;
2  /**
3   * MahasiswaMain04
4   */
5  public class MahasiswaMain04 {
6
7      public static void main(String[] args) {
8          Scanner s = new Scanner(System.in);
9          Scanner sl = new Scanner(System.in);
10
11          int jumMhs = 5;
12
13          System.out.println("Input jumlah Mahasiswa: ");
14          int maxSize = s.nextInt();
15
16          PencarianMhs04 data = new PencarianMhs04(maxSize);
17
18          System.out.println("-----");
19          System.out.println("Masukkan data mahasiswa secara urut dari NIM terkecil : ");
20          for (int i = 0; i < maxSize; i++) {
21              System.out.println("-----");
22              System.out.print("Nim\t : ");
23              int nim = s.nextInt();
24              System.out.print("Nama\t : ");
25              String nama = sl.nextLine();
26              System.out.print("Umur\t : ");
27              int umur = s.nextInt();
28              System.out.print("IPK\t : ");
29              double ipk = s.nextDouble();
30
31              Mahasiswa04 m = new Mahasiswa04(nama, nim, umur, ipk);
32              data.tambah(m);
33          }
34          System.out.println("-----");
35          System.out.println("Data Keseluruhan Mahasiswa : ");
36          data.tampil();
37
38          System.out.println("-----");
39          System.out.println("Pencarian data : ");
40          System.out.println("Masukkan NIM Mahasiswa yang dicari: ");
41          System.out.print("NIM : ");
42          int cari = s.nextInt();
43          System.out.println("Menggunakan sequential Search");
44          int posisi = data.FindSeqSearch(cari);
45          data.TampilPosisi(cari, posisi);
46          data.TampilData(cari, posisi);
47
48          System.out.println("=====");
49          System.out.println("Menggunakan binary Search");
50          posisi = data.FindBinarySearch(cari, 0, maxSize - 1);
51          data.TampilPosisi(cari, posisi);
52          data.TampilData(cari, posisi);
53
54      }
55
56  }

```

Berikut merupakan output dari kode program di atas

```
Input jumlah Mahasiswa:
3
-----
Masukkan data mahasiswa secara urut dari NIM terkecil :
-----
Nim      : 2021
Nama     : dygta
Umur     : 21
IPK      : 3.7
-----
Nim      : 2022
Nama     : nasya
Umur     : 20
IPK      : 4.0
-----
Nim      : 2023
Nama     : nisa
Umur     : 19
IPK      : 3.9
-----

Data Keseluruhan Mahasiswa :
Nim = 2021
Nama = dygta
Umur = 21
IPK = 3.7
-----
Nim = 2022
Nama = nasya
Umur = 20
IPK = 4.0
-----
Nim = 2023
Nama = nisa
Umur = 19
IPK = 3.9
-----

-----
Pencarian data :
Masukkan NIM Mahasiswa yang dicari:
NIM : 2022
Menggunakan sequential Search
data : 2022
ditemukan pada indeks 2
Nim      : 2022
Nama     : nasya
Umur     : 20
IPK      : 4.0
=====
Menggunakan binary Search
data : 2022
ditemukan pada indeks 2
Nim      : 2022
Nama     : nasya
Umur     : 20
IPK      : 4.0
C:\Users\user\Documents\ASD Jobsheet 7>
```

## 6.4. Percobaan Pengayaan Divide and Conquer

### 6.4.1. Langkah-langkah Percobaan Merge Sort

```
1  /**
2   * MergeSorting04
3   */
4  public class MergeSorting04 {
5
6      public void mergeSort(int[] data) {
7          sort(data, 0, data.length - 1);
8      }
9
10     public void merge (int data[], int left, int middle, int right) {
11         int[] temp = new int[data.length];
12         for (int i = left; i <= right; i++) {
13             temp[i] = data[i];
14         }
15         int a = left;
16         int b = middle + 1;
17         int c = left;
18
19         while (a <= middle && b <= right) {
20             if (temp[a] <= temp[b]) {
21                 data[c] = temp[a];
22                 a++;
23             } else {
24                 data[c] = temp[b];
25                 b++;
26             }
27             c++;
28         }
29
30         while (a <= middle) {
31             data[c] = temp[a];
32             a++;
33             c++;
34         }
35     }
36
37     public void sort (int data[], int left, int right) {
38         if (left < right) {
39             int middle = (left + right) / 2;
40             sort(data, left, middle);
41             sort(data, middle + 1, right);
42             merge(data, left, middle, right);
43         }
44     }
45
46     public void printArray(int arr[]){
47         int n = arr.length;
48         for (int i = 0; i < n; i++) {
49             System.out.println(arr[i] + " ");
50         }
51         System.out.println();
52     }
53 }
```

```

1  public class SortMain04 {
2      public static void main(String[] args) {
3          int data[] = {10,40,30,50,70,20,100,90};
4          System.out.println("Sorting dengan merge sort");
5          MergeSorting04 mSort = new MergeSorting04();
6          System.out.println("data awal");
7          mSort.printArray(data);
8          mSort.mergeSort(data);
9          System.out.println("setelah di urutkan");
10         mSort.printArray(data);
11     }
12 }

```

#### 6.4.2. Verifikasi Hasil Percobaan

```

Sorting dengan merge sort
data awal
10
40
30
50
70
20
100
90

setelah di urutkan
10
20
30
40
50
70
90
100

C:\Users\user\Documents\ASD Jobsheet 7>

```

#### 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!

```

1  /**
2   * Mahasiswa04
3   */
4  public class Mahasiswa04 {
5
6      int nim;
7      String nama;
8      int umur;
9      double ipk;
10
11      Mahasiswa04 (int ni, String n, int u, double i) {
12
13          nim =ni;
14          nama =n;
15          umur =u;
16          ipk =i;
17      }
18      void tampil() {
19          System.out.println("Nim = " + nim);
20          System.out.println("Nama = " + nama);
21          System.out.println("Umur = " + umur );
22          System.out.println("IPK = " + ipk);
23      }
24  }

```

```

1  import java.util.Scanner;
2  /**
3   * MahasiswaMain04
4   */
5  public class MahasiswaMain04 {
6
7      public static void main(String[] args) {
8          Scanner s = new Scanner(System.in);
9          Scanner s1 = new Scanner(System.in);
10
11          int jumMhs = 5;
12
13          System.out.println("Input jumlah Mahasiswa: ");
14          int maxSize = s.nextInt();
15
16          PencarianMhs04 data = new PencarianMhs04(maxSize);
17
18          System.out.println("-----");
19          System.out.println("Masukkan data mahasiswa secaraurut dari NIM terkecil : ");
20          for (int i = 0; i < maxSize; i++) {
21              System.out.println("-----");
22              System.out.print("Nim\t : ");
23              int nim = s.nextInt();
24              System.out.print("Nama\t : ");
25              String nama = s1.nextLine();
26              System.out.print("Umur\t : ");
27              int umur = s.nextInt();
28              System.out.print("IPK\t : ");
29              double ipk = s.nextDouble();
30
31              Mahasiswa04 m = new Mahasiswa04(nim, nama, umur, ipk);
32              data.tambah(m);
33          }
34          System.out.println("-----");
35          System.out.println("Data Keseluruhan Mahasiswa : ");
36          data.tampil();
37
38          System.out.println("-----");
39          System.out.println("Pencarian data : ");
40          System.out.println("Masukkan Nama Mahasiswa yang dicari: ");
41          System.out.print("Nama: ");
42          String cari = s.next();
43
44          System.out.println("=====");
45          System.out.println("Menggunakan Binary search");
46          int posisi = data.FindBinarySearch(cari, 0, maxSize - 1);
47          data.TampilPosisi(cari, posisi);
48          data.TampilData(cari, posisi);
49      }
50  }

```



```

1  /**
2   * PencarianMhs04
3   */
4  public class PencarianMhs04 {
5      Mahasiswa04 listMhs[];
6      int idx;
7
8      public PencarianMhs04(int maxSize) {
9          listMhs = new Mahasiswa04[maxSize];
10     }
11     void tambah(Mahasiswa04 m) {
12         if(idx < listMhs.length) {
13             listMhs[idx] = m;
14             idx++;
15         } else {
16             System.out.println("Data sudah penuh!!");
17         }
18     }
19
20     void tampil() {
21         for (int i = 0; i < idx; i++) {
22             listMhs[i].tampil();
23             System.out.println("-----");
24         }
25     }
26
27     public int FindSeqSearch (int cari) {
28         int posisi = -1;
29         for (int j = 0; j < idx; j++) {
30             if (listMhs[j].nim == cari) {
31                 posisi = j;
32                 break;
33             }
34         }
35         return posisi;
36     }
37
38     public int FindBinarySearch(String cari, int left, int right) {
39         int mid;
40         if (right >= left) {
41             mid = (left + right) / 2;
42             int compareResult = cari.compareTo(listMhs[mid].nama);
43             if (compareResult == 0) {
44                 return mid;
45             } else if (compareResult < 0) {
46                 return FindBinarySearch(cari, left, mid - 1);
47             } else {
48                 return FindBinarySearch(cari, left, right);
49             }
50         }
51         return -1;
52     }
53     public void TampilPosisi (String cari, int pos) {
54         if (pos != -1) {
55             System.out.println("Data " + cari + "ditemukan pada indeks " + (
56 pos+1));
57         } else {
58             System.out.println("Data " + cari + "tidak ditemukan");
59         }
60     }
61
62     public void TampilData (String cari, int pos) {
63         if (pos != -1) {
64             int startIdx = pos;
65             int endIdx = pos;
66
67             while (startIdx >= 0 && listMhs[endIdx].nama.equals(cari)) {
68                 startIdx--;
69             }
70             startIdx++;
71
72             while (endIdx < idx && listMhs[endIdx].nama.equals(cari)){
73                 endIdx++;
74             }
75             endIdx--;
76
77             if (startIdx != endIdx) {
78                 System.out.println("Peringatan: lebih dari satu mahasiswa dengan nama " + cari + "ditemukan");
79             }
80             for(int i = startIdx; i <= endIdx; i++) {
81                 System.out.println("Nim\t : " + listMhs[pos].nim);
82                 System.out.println("Nama\t : " + cari);
83                 System.out.println("Umur\t : " + listMhs[pos].umur);
84                 System.out.println("IPK\t : " + listMhs[pos].ipk);
85                 System.out.println("-----");
86             }
87         } else {
88             System.out.println("data " + cari + "tidak ditemukan");
89         }
90     }
91 }
92 }

```

Berikut merupakan output dari kode program di atas

```
Input jumlah Mahasiswa:
3
-----
Masukkan data mahasiswa secara urut dari NIM terkecil :
-----
Nim      : 2019
Nama     : nasya
Umur     : 21
IPK      : 3.8
-----
Nim      : 2020
Nama     : annisa
Umur     : 20
IPK      : 3.9
-----
Nim      : 2021
Nama     : annisa
Umur     : 19
IPK      : 4
-----
```

```
Data Keseluruhan Mahasiswa :
Nim = 2019
Nama = nasya
Umur = 21
IPK = 3.8
-----
Nim = 2020
Nama = annisa
Umur = 20
IPK = 3.9
-----
Nim = 2021
Nama = annisa
Umur = 19
IPK = 4.0
-----
```

```
Pencarian data :
Masukkan Nama Mahasiswa yang dicari:
Nama: annisa
=====
Menggunakan Binary search
Data annisaditemukan pada indeks 2
Peringatan: lebih dari satu mahasiswa dengan nama annisaditemukan
Nim      : 2020
Nama     : annisa
Umur     : 20
IPK      : 3.9
=====
Nim      : 2020
Nama     : annisa
Umur     : 20
IPK      : 3.9
=====
Nim      : 2020
Nama     : annisa
Umur     : 20
IPK      : 3.9
=====
C:\Users\user\Documents\ASD Jobsheet 7>
```