LAPORAN PRAKTIKUM
MATA KULIAH ALGORITMA DAN STRUKTUR DATA

PERTEMUAN 7 : SEARCHING

KAYLA RACHMAUDINA SATITI PUTRI

2341760103

D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG

2024

JOBSHEET VI
SEARCHING

6.1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Menjelaskan mengenai algoritma Searching.
2. Membuat dan mendeklarasikan struktur algoritma Searching.
3. Menerapkan dan mengimplementasikan algoritma Searching.

6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

6.2.1 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 array of integer

```java
public class sorting15 {
    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);
            }
        }

        System.out.println("Data tidak ditemukan");
    }
```

3. Tambahkan fungsi main sebagai berikut

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

4. Compile dan run program

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

6.2.2 Sequential Search Menggunakan Array of Object

Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa.

Mahasiswa
Nim: int nama: String umur: int ipk: double
Mahasiswa(ni:int, n: String, u: int, i: double) tampil(): void

Berdasarkan class diagram di atas, akan dibuat class Mahasiswa yang berfunsi untuk membuat objek mahasiswa yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga fungsi tampil() untuk menampilkan semua attribute yang ada.

PencarianMhs
listMhs: Mahasiswa[5] idx: int
tambah(mhs: Mahasiswa): void tampil(): void FindSeqSearch(int cari): int Tampilposisi(int x,int pos): void TampilData(int x,int pos) :void

Selanjutnya class diagram di atas merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array mahasiswa, misalkan untuk menambahkan objek mahasiswa, menampilkan semua data mahasiswa, untuk melakukan pencarian berdasarkan NIM menggunakan algoritma Sequential Search, menampilkan posisi dari data yang dicari, serta menampilkan data mahasiswa yang dicari.

✞ Langkah-langkah Percobaan Sequential Search

1. Buatlah Project baru pada Netbeans dengan nama **`TestSearching`**
2. Kemuadian buat packages baru dengan nama **`minggu7`**.
3. Buat class **`Mahasiswa`**, kemudian deklarasikan atribut berikut ini:

```java
public class Mahasiswa15 {
    int nim;
    String nama;
    int umur;
    double ipk;
```

4. Buatlah konstruktor dengan nama **`Mahasiswa`** dengan parameter (**`int ni, String n, int u, double i`**)

```java
    Mahasiswa15(int ni, String n, int u, double i) {
        nim = ni;
        nama = n;
        umur = u;
        ipk = i;
    }
```

5. Buatlah method **`tampil`** bertipe void.

```java
    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`**

```java
public class PencarianMhs15 {
    Mahasiswa15 listMhs[] = new Mahasiswa15[5];
    int idx;
```

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

```
5       void tambah(Mahasiswa15 m) {
6           if (idx < listMhs.length) {
7               listMhs[idx] = m;
8               idx++;
9           } else {
10              System.out.println("Data sudah penuh!!");
11          }
12      }
```

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.

```
14      void tampil() {
15          for (Mahasiswa15 m : listMhs) {
16              m.tampil();
17              System.out.println("------------------------------");
18          }
19      }
```

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

```
21      public int FindSeqSearch(int cari) {
22          int posisi = -1;
23          for (int j = 0; j < listMhs.length; j++) {
24              if (listMhs[j].nim == cari) {
25                  posisi = j;
26                  break;
27              }
28          }
29          return posisi;
30      }
```

10. Buatlah method **`Tampilposisisi`** bertipe void dan Deklarasikan isi dari method

```
32      public void TampilPosisi(int x, int pos) {
33          if (pos != -1) {
34              System.out.println("Data : " + x + " ditemukan pada index " + pos);
35          } else {
36              System.out.println("Data : " + x + " tidak ditemukan");
37          }
38      }
```

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

```
40      public void TampilData(int x, int pos) {
41          if (pos != -1) {
42              System.out.println("Nim\t : " + x);
43              System.out.println("Nama\t : " + listMhs[pos].nama);
44              System.out.println("Umur\t : " + listMhs[pos].umur);
45              System.out.println("IPK\t : " + listMhs[pos].ipk);
46          } else {
47              System.out.println("Data : " + x + " tidak ditemukan");
48          }
49      }
50  }
```

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

```
1  import java.util.Scanner;
2
3  public class MahasiswaMain15 {
4      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.

```
5           Scanner s = new Scanner(System.in);
6           Scanner s1 = new Scanner(System.in);
7
8           PencarianMhs15 data = new PencarianMhs15();
9           int jumMhs = 5;
10
11          System.out.println("-----------------------------------------");
12          System.out.println("Masukkan data mahasiswa secara urut dari nim terkecil");
13          for (int i = 0; i < jumMhs; i++) {
14              System.out.println("----------------------------");
15              System.out.print("Nim\t: ");
16              int nim = s.nextInt();
17              System.out.print("Nama\t: ");
18              String nama = s1.nextLine();
19              System.out.print("Umur\t: ");
20              int umur = s.nextInt();
21              System.out.print("Nim\t: ");
22              double ipk = s.nextDouble();
23
24              Mahasiswa15 m = new Mahasiswa15(nim, nama, umur, ipk);
25              data.tambah(m);
26          }
```

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

```
28              System.out.println("-----------------------------------------");
29              System.out.println("Data keseluruhan mahasiswa : ");
30              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.

```
32              System.out.println("_____________________________________________");
33              System.out.println("_____________________________________________");
34              System.out.println("Pencarian data : ");
35              System.out.println("Masukkan nim mahasiswa yang dicari : ");
36              System.out.println("NIM : ");
37              int cari = s.nextInt();
38              System.out.println("Menggunakan sequential search");
39              int posisi = data.FindSeqSearch(cari);
```

16. Lakukan pemanggilan method **`Tampilposisi`** dari class **`PencarianMhs.`**

```
41              data.TampilPosisi(cari, posisi);
```

17. Lakukan pemanggilan method **`TampilData`** dari class **`PencarianMhs.`**

```
42              data.TampilData(cari, posisi);
43      }
```

18. Jalankan dan amati hasilnya.

6.2.2. Verifikasi Hasil Percobaan

```
-----------------------------------------
Masukkan data mahasiswa secara urut dari nim terkecil
-----------------------------
Nim     : 2017
Nama    : dewi lestari
Umur    : 23
Nim     : 3.5
-----------------------------
Nim     : 2018
Nama    : sinta sanjaya
Umur    : 22
Nim     : 4
-----------------------------
Nim     : 2019
Nama    : danang adi
Umur    : 22
Nim     : 3.7
-----------------------------
Nim     : 2020
Nama    : budi prakoso
Umur    : 20
Nim     : 2.9
-----------------------------
Nim     : 2021
Nama    : vania siti
Umur    : 20
Nim     : 3
```

```
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 prakoso
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 index 1
Nim      : 2018
Nama     : sinta sanjaya
Umur     : 22
IPK      : 4.0
```

6.2.3. Pertanyaan

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

```
public static void main(String[] args) {
    int[] daftarNilai = {10, 5, 20, 15, 5, 45};
    sorting15.sequentialSearch(daftarNilai, key:5);
```

2. Jelaskan perbedaan method **`TampilData`** dan **`Tampilposisi`** pada class PencarianMhs

   TampilData menampilkan informasi lengkap data, sedangkan TampilPosisi hanya menunjukkan posisinya di array.

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

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

Fungsi break pada potongan kode tersebut digunakan untuk menghentikan loop for setelah data yang dicari dengan NIM cari ditemukan.

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 dengan data yang tidak terurut, namun hasil yang dikeluarkan bisa jadi tidak akurat untuk method TampilData. Hal ini terjadi karena method TampilData bergantung pada urutan data yang ditemukan oleh FindSeqSearch, sedangkan data tidak terurut.

```
Data keseluruhan mahasiswa :
Nim   : 2021
Nama  : vania siti
Umur  : 20
IPK   : 3.0
------------------------------
Nim   : 2020
Nama  : budi prakoso
Umur  : 20
IPK   : 2.9
------------------------------
Nim   : 2016
Nama  : danang adi
Umur  : 22
IPK   : 3.7
------------------------------
Nim   : 2024
Nama  : sinta sanjaya
Umur  : 22
IPK   : 4.0
------------------------------
Nim   : 2000
Nama  : sinta sanjaya
Umur  : 23
IPK   : 3.0
------------------------------

_____________________________________________
_____________________________________________
Pencarian data :
Masukkan nim mahasiswa yang dicari :
NIM :
2020
Menggunakan sequential search
Data : 2020 ditemukan pada index 1
Nim      : 2020
Nama     : budi prakoso
Umur     : 20
IPK      : 2.9
```

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

```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()

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

    int[] sortedNilai = {5, 5, 10, 20, 30, 40, 50};
    int index = binarySearchAsc(sortedNilai, key:5);

    if (index != -1) {
        System.out.println("Data ditemukan pada index ke-" + index);
    } else {
        System.out.println(x:"Data tidak ditemukan");
    }
}
```

3. Run dan compile program

```
Data ditemukan pada indeks ke-5
Data tidak ditemukan
Data ditemukan pada index ke-1
```

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.

```
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 **`Mahasiswamain.`** Kemudia 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);
```

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    : danang adi
Umur    : 22
IPK     : 3.7
-----------------------------
Nim     : 2020
Nama    : budi prakoso
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 prakoso
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 index 1
Nim      : 2018
Nama     : sinta sanjaya
Umur     : 22
IPK      : 4.0
========================================
Menggunakan binary search
Data     : 2018 ditemukan pada index 1
Nim      : 2018
Nama     : sinta sanjaya
Umur     : 22
IPK      : 4.0
```

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

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

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

```
    return FindBinarySearch(cari, left, mid - 1);
} else {
    return FindBinarySearch(cari, mid + 1, right);
```

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

   Program masih dapat berjalan dengan data yang tidak terurut, namun hasil yang dikeluarkan bisa jadi tidak akurat. Hal ini dikarenakan metode pencarian binary membutuhkan data yang telah berurutan sebelumnya agar dapat bekerja dengan benar.

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

```
========================================
Menggunakan binary search
Data     : 20210 tidak ditemukan
Data     : 20210 tidak ditemukan
```

```
-----------------------------------------
-----------------------------------------
Pencarian data :
Masukkan nim mahasiswa yang dicari :
NIM :
35
=========================================
Menggunakan sequential search
Data     : 35 ditemukan pada index 2
Nim      : 35
Nama     : y
Umur     : 18
IPK      : 3.7
=========================================
Menggunakan binary search
Data     : 35 ditemukan pada index 2
Nim      : 35
Nama     : y
Umur     : 18
IPK      : 3.7
```

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

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

```java
public PencarianMhs15(int jumMhs) {
    listMhs = new Mahasiswa15[jumMhs];
}
```

```
System.out.println("------------------------------------------------------------");
System.out.print("Masukkan berapa banyak data yang ingin ditambahkan : ");
jumMhs = s.nextInt();
```

6.4. Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

1 Buatlah Package baru pada NetBeans dengan nama **`MergeSortTest`**

7 Tambahkan class **`MergeSorting`** pada package tersebut

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

```
public class MergeSorting15 {
    public void mergeSort(int[] data) {
```

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

```
    private void merge(int data[], int left, int middle, int right) {
```

10. Implementasikan proses merge sebagai berikut.

```
    private 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];
        }
    }
```

11. Buatlah method sort

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

12. Implementasikan kode berikut pada method sort

```
    private void sort(int data[], int left, int right ) {
        // membagi jd 2 bagian dan dibagi lg smp g bs dibagi
        if (left < right) {
            int middle = (left + right) / 2;
            sort(data, left, middle);
            sort(data, middle + 1, right);
            merge(data, left, middle, right);
        }
    }
```

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

```
    public void mergeSort(int[] data) {
        sort(data, 0, data.length - 1);
    }
```

14 Tambahkan method **`printArray`**

```
42      public void printArray(int arr[]) {
43          int n = arr.length;
44          for (int i = 0; i < n; i++) {
45              System.out.print(arr[i] + " ");
46          }
47          System.out.println();
48      }
```

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

```
1  public class SortMain15 {
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          MergeSorting15 mSort = new MergeSorting15();
6          System.out.println("Data awal");
7          mSort.printArray(data);
8          mSort.mergeSort(data);
9          System.out.println("Setelah diurutkan");
10         mSort.printArray(data);
11     }
12 }
```

6.4.2. 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
```

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!

```
System.out.println(x:"_____");
System.out.println(x:"_____");
System.out.println(x:"Pencarian data : ");
System.out.print(s:"Masukkan nama mahasiswa yang dicari : ");
String cari = s.next();
```

```
System.out.println(x:"============================================================");
System.out.println(x:"Menggunakan binary search");

data.countSearchName(cari);
int posisi = data.FindBinarySearchByName(cari, left:0, jumMhs - 1, count:0);
data.TampilposisiNama(cari, posisi);
data.TampilDataNama (cari, posisi);
```

```java
public void TampilPosisi(int x, int pos) {
    if (pos != -1) {
        System.out.println("Data\t : " + x + " ditemukan pada index " + pos);
    } else {
        System.out.println("Data\t : " + x + " tidak ditemukan");
    }
}

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\t : " + x + " tidak ditemukan");
    }
}

public void TampilposisiNama(String x, int pos) {
    if (pos != -1) {
        System.out.println("Data\t : " + x + " ditemukan pada indeks " + pos);
    } else {
        System.out.println("Data\t " + x + " tidak ditemukan");
    }
}

public void TampilDataNama(String 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\t " + x + " tidak ditemukan");
    }
}

public int FindBinarySearchByName(String cari, int left, int right, int count) {
    if (right >= left) {

        int mid = (left + right) / 2;
        int comparison = listMhs[mid].nama.compareTo(cari);

        if (comparison < 0) {
            return FindBinarySearchByName(cari, mid + 1, right, count);
        } else if (comparison > 0) {
            return FindBinarySearchByName(cari, left, mid - 1, count);
        } else {
            return mid;
        }
    }
    return -1;
}

public void countSearchName(String cari) {
    int count = 0;
    for(int i=0; i<listMhs.length; i++){
        if(cari.equalsIgnoreCase(listMhs[i].nama)){
            count += 1;
        }
    }

    if (count > 1) {
        System.out.println("Data yang ditemukan lebih dari 1");
        return;
    }
}
```

OUTPUT

```
============================================================
Pencarian data :
Masukkan nama mahasiswa yang dicari :
c
============================================================
Menggunakan binary search
Data     : c ditemukan pada indeks 2
Nim      : c
Nama     : c
Umur     : 2
IPK      : 3.8
```

```
============================================================
Pencarian data :
Masukkan nama mahasiswa yang dicari :
kay
============================================================
Menggunakan binary search
Data yang ditemukan lebih dari 1
```