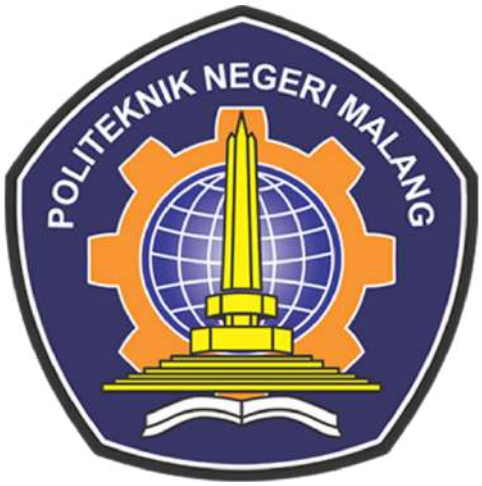
LAPORAN PRAKTIKUM
MATA KULIAH ALGORITMA DAN STRUKTUR DATA

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

PERTEMUAN - 7 - Searching



Nama : **M. Zidna Billah Faza**

NIM : **2341760030**

Prodi : **D-IV Sistem Informasi Bisnis**

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

Percobaan 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 static void SequentialSearch(int[] arr, int key) {

    for (int i = 0; i < arr.length; i++) {
        if (i == key) {
            System.out.println("Data ditemuukan pada indeks ke-" + i);
        }
    }

    System.out.println(x:"Data tidak ditemukan");

}
```

3) Tambahkan fungsi main sebagai berikut

```java
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

```
Data ditemuukan pada indeks ke-5
Data tidak ditemukan
```

Langkah - langkah Sequential Search Menggunakan Array of Object

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

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

```
public class Mahasiswa_18 {

    int nim;
    String nama;
    int umur;
    double ipk;
```

5) Buatlah method tampil bertipe void

```
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 PencarianMahasiswa_18 {

    Mahasiswa_18 listMahasiswa[] = new Mahasiswa_18[5];
    int index;
```

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

```
void Tambah(Mahasiswa_18 m) {

    if (index < listMahasiswa.length) {

        listMahasiswa[index] = m;
        index++;

    } 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.

```
void Tampil() {

    for (Mahasiswa_18 m : listMahasiswa) {

        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 FindSequentialSearch(int cari) {

    int posisi = -1;

    for (int j = 0; j < listMahasiswa.length; j++) {

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

    }

    return posisi;

}
```

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

```
public void TampilPosisi(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 : " + listMahasiswa[pos].nama);
        System.out.println("Umur\t : " + listMahasiswa[pos].umur);
        System.out.println("IPK\t : " + listMahasiswa[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_18 {
    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 input18 = new Scanner(System.in);
Scanner input18Line = new Scanner(System.in);

PencarianMahasiswa_18 data = new PencarianMahasiswa_18();
int jumlahMahasiswa = 5;

System.out.println(x:"============================================================");
System.out.println(x:"   Masukkan data mahasiswa secara urut dari NIM terkecil");
System.out.println(x:"============================================================");

for (int i = 0; i < jumlahMahasiswa; i++) {

    System.out.print(s:"NIM\t : ");
    int nim = input18.nextInt();
    System.out.print(s:"Nama\t : ");
    String nama = input18Line.nextLine();
    System.out.print(s:"Umur\t : ");
    int umur = input18.nextInt();
    System.out.print(s:"IPK\t : ");
    double ipk = input18.nextDouble();
    System.out.println(x:"============================================================");

    Mahasiswa_18 m = new Mahasiswa_18(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");
System.out.println(x:"============================================================");
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:"                  Pencarian data mahasiswa");
System.out.println(x:"============================================================");
System.out.print(s:"Masukkan NIM mahasiswa yang ingin dicari : ");
int cari = input18.nextInt();

System.out.println(x:"============================================================");
System.out.println(x:"              Menggunakan Sequential Search");
System.out.println(x:"============================================================");
int posisi = data.FindSequentialSearch(cari);
```

16) Lakukan pemanggilan method Tampilposisi dari class PencarianMhs.

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

17) Lakukan pemanggilan method TampilData dari class PencarianMhs.

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

18) Jalankan dan amati hasilnya serta 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 mahasiswa
============================================================
Masukkan NIM mahasiswa yang ingin dicari : 2018
============================================================
              Menggunakan Sequential Search
============================================================
Data 2018 ditemukan pada indeks 1
NIM      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
============================================================
```

Pertanyaan Percobaan 1

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

```
int[] daftarNilai = { 10, 5, 20, 15, 5, 45 };
SequentialSearch(daftarNilai, key:5);
```

```
Data ditemuukan pada indeks ke-5
Data tidak ditemukan
```

2) Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs

Pada method TampilData berfungsi untuk menampilkan seluruh data Mahasiswa beserta atributnya sedangkan TampilPosisi berfungsi untuk mengetahui index dari data Mahasiswa yang dicari

3) Jelaskan fungsi break pada kode program dibawah ini!

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

Fungsi break pada kode program tersebut adalah untuk menghentikan looping sehingga ketika listMahasiswa[j].nim == cari maka posisi = j dan program keluar looping

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 akan jalan dengan normal karena pada Sequential Search membandingan data urut dari depan ke belakang dan data yang ada tidak harus urut.

Percobaan 2 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()

```
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

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

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 == listMahasiswa[mid].nim) {
            return mid;
        } else if (listMahasiswa[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");
System.out.println(x:"============================================================");
posisi = data.FindBinarySearch(cari, left:0, jumlahMahasiswa - 1);

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

3) Jalankan dan amati hasilnya serta 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 mahasiswa
=============================================================
Masukkan NIM mahasiswa yang ingin dicari : 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
```

Pertanyaan Percobaan 2

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

```
if (right >= left) {
    mid = (left + right) / 2;
```

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

```
if (cari == listMahasiswa[mid].nim) {
    return mid;
} else if (listMahasiswa[mid].nim > cari) {
    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 akan menerima inputan dengan normal namun ketika melakukan pencarian BinarySearch maka data yang dicari tidak dapat ditemukan karena pada BinarySearch data yang dicari harus urut terlebih dahulu.

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

Ketika akan mengurutkan data dari terbesar ke terkecil maka harus dilakukan modifikasi pada tanda lebih dari (>) agar program dapat berjalan sesuai dengan perintah

```
if (cari == listMahasiswa[mid].nim) {
    return mid;
} else if (listMahasiswa[mid].nim < cari) { // Modifikasi pada tanda lebih dari (>) ke (<)
    return FindBinarySearch(cari, left, mid -1);
} else {
    return FindBinarySearch(cari, mid + 1, right);
}
```

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

```
System.out.println(x:"============================================================");
System.out.print(s:"Masukkan jumlah mahasiswa    : ");
int jumlahMahasiswa = input18.nextInt();
PencarianMahasiswa_18 data = new PencarianMahasiswa_18(jumlahMahasiswa);
System.out.println(x:"============================================================");
```

```
Mahasiswa_18 listMahasiswa[];
int index;

public PencarianMahasiswa_18(int jumlahMahasiswa) {
    listMahasiswa = new Mahasiswa_18[jumlahMahasiswa];
    index = 0;
}
```

Percobaan 3 Pengayaan Divide and Conquer

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.

```
public void Merge(int data[], int left, int middle, 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;

    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

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

7) Implementasikan kode berikut pada method sort

```
public 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

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

    Sort(data, left:0, data.length -1);

}
```

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_18 {

    Run | Debug
    public static void main(String[] args) {
        int data[] = { 10, 40, 30, 50, 70, 20, 100, 90 };

        System.out.println(x:"=======================================");
        System.out.println(x:"        Sorting dengan merge sort");
        System.out.println(x:"=======================================");
        MergeSorting_18 mSort = new MergeSorting_18();

        System.out.println(x:"              DATA AWAL");
        System.out.println(x:"=======================================");
        mSort.PrintArray(data);
        mSort.mergeSort(data);

        System.out.println(x:"=======================================");
        System.out.println(x:"         Data setelah diurutkan");
        System.out.println(x:"=======================================");
        mSort.PrintArray(data);
        System.out.println(x:"=======================================");
    }

}
```

11) Verifikasi hasil percobaan

```
=======================================
        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
=======================================
```

Latihan Praktikum

Modifikasi percobaan searching diatas yang menggunakan Searching array of object dengan ketentuan berikut ini :

1) Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma binary Search)
2) Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan!

Jawaban

- **Program Mahasiswa_18.java**

```java
public class Mahasiswa_18 {

    int nim;
    String nama;
    int umur;
    double ipk;

    Mahasiswa_18(int nim, String nama, int umur, double ipk) {

        this.nim = nim;
        this.nama = nama;
        this.umur = umur;
        this.ipk = ipk;

    }

    void Tampil() {

        System.out.println("NIM     : " + nim);
        System.out.println("Nama    : " + nama);
        System.out.println("Umur    : " + umur);
        System.out.println("IPK     : " + ipk);

    }

}
```

- **Program PencarianMahasiswa_18_1.java**

```java
import java.util.Arrays;

public class PencarianMahasiswa_18_1 {

    Mahasiswa_18 listMahasiswa[];
    int index;

    public PencarianMahasiswa_18_1(int jumlahMahasiswa) {
        listMahasiswa = new Mahasiswa_18[jumlahMahasiswa];
        index = 0;
    }

    void Tambah(Mahasiswa_18 m) {
        if (index < listMahasiswa.length) {
            listMahasiswa[index] = m;
            index++;
        } else {
            System.out.println("Data sudah penuh !!!");
        }
    }

    void Tampil() {
        for (Mahasiswa_18 m : listMahasiswa) {
            m.Tampil();
            System.out.println("============================================================");
        }
    }

    // Metode untuk melakukan pencarian nama mahasiswa menggunakan binary search
    public int FindBinarySearch(String cari, int left, int right) {
        int mid;
        while (left <= right) {
            mid = left + (right - left) / 2;
            int result = cari.compareTo(listMahasiswa[mid].nama);
            if (result == 0) {
                // Jika nama ditemukan, lanjutkan pencarian ke kiri dan kanan untuk mencari hasil tambahan
                int count = 1;
                int leftIndex = mid - 1;
                int rightIndex = mid + 1;

                // Cari ke kiri
                while (leftIndex >= 0 && listMahasiswa[leftIndex].nama.equals(cari)) {
                    count++;
                    leftIndex--;
                }

                // Cari ke kanan
                while (rightIndex < listMahasiswa.length && listMahasiswa[rightIndex].nama.equals(cari)) {
                    count++;
                    rightIndex++;
                }

                System.out.println("Ditemukan " + count + " hasil dengan nama yang sama.");
                return mid; // Kembalikan indeks pertama yang ditemukan
            } else if (result < 0) {
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }
        return -1; // Jika tidak ditemukan
    }

    public void TampilPosisi(int pos) {
        if (pos != -1) {
            System.out.println("Data ditemukan pada indeks " + pos);
        } else {
            System.out.println("Data tidak ditemukan");
        }
    }

    public void TampilData(int pos) {
        if (pos != -1) {
            listMahasiswa[pos].Tampil();
        } else {
            System.out.println("Data tidak ditemukan");
        }
    }
}

```

- **Program MahasiswaMain_18_1.java**

```java
import java.util.Scanner;

public class MahasiswaMain_18_1 {
    public static void main(String[] args) {

        Scanner input18 = new Scanner(System.in);
        Scanner input18Line = new Scanner(System.in);

        System.out.println("============================================================");
        System.out.print("Masukkan jumlah mahasiswa   : ");
        int jumlahMahasiswa = input18.nextInt();
        System.out.println("============================================================");

        PencarianMahasiswa_18_1 data = new PencarianMahasiswa_18_1(jumlahMahasiswa);

        System.out.println("============================================================");
        System.out.println("                  Masukkan data mahasiswa");
        System.out.println("============================================================");

        for (int i = 0; i < jumlahMahasiswa; i++) {

            System.out.print("Nama\t : ");
            String nama = input18Line.nextLine();
            System.out.print("NIM\t : ");
            int nim = input18.nextInt();
            System.out.print("Umur\t : ");
            int umur = input18.nextInt();
            System.out.print("IPK\t : ");
            double ipk = input18.nextDouble();
            System.out.println("============================================================");

            Mahasiswa_18 m = new Mahasiswa_18(nim, nama, umur, ipk);
            data.Tambah(m);

        }

        System.out.println("============================================================");
        System.out.println("                Data keseluruhan mahasiswa");
        System.out.println("============================================================");
        data.Tampil();

        System.out.println("============================================================");
        System.out.println("                  Pencarian data mahasiswa");
        System.out.println("============================================================");
        System.out.print("Masukkan Nama mahasiswa yang ingin dicari : ");
        String cariNama = input18Line.nextLine();

        System.out.println("============================================================");
        System.out.println("              Menggunakan Binary Search");
        System.out.println("============================================================");
        int posisi = data.FindBinarySearch(cariNama, 0, jumlahMahasiswa - 1);

        data.TampilPosisi(posisi);
        data.TampilData(posisi);
    }
}

```

- **Output**

```
============================================================
Masukkan jumlah mahasiswa   : 3
============================================================
============================================================
                  Masukkan data mahasiswa
============================================================
Nama     : Steve Rogers
NIM      : 63231
Umur     : 25
IPK      : 3.8
============================================================
Nama     : Thor Oddinson
NIM      : 32413
Umur     : 22
IPK      : 3.7
============================================================
Nama     : Loki Laufey
NIM      : 82341
Umur     : 24
IPK      : 3.5
============================================================
============================================================
                Data keseluruhan mahasiswa
============================================================
NIM     : 63231
Nama    : Steve Rogers
Umur    : 25
IPK     : 3.8
============================================================
NIM     : 32413
Nama    : Thor Oddinson
Umur    : 22
IPK     : 3.7
============================================================
NIM     : 82341
Nama    : Loki Laufey
Umur    : 24
IPK     : 3.5
============================================================
============================================================
                  Pencarian data mahasiswa
============================================================
Masukkan Nama mahasiswa yang ingin dicari : Thor Oddinson
============================================================
               Menggunakan Binary Search
============================================================
Ditemukan 1 hasil dengan nama yang sama.
Data ditemukan pada indeks 1
NIM     : 32413
Nama    : Thor Oddinson
Umur    : 22
IPK     : 3.7
```

Link Repository : https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data