

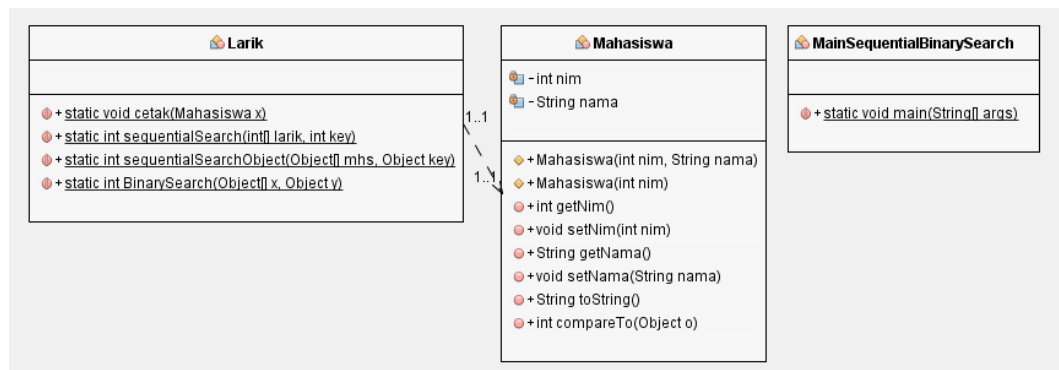
LAPORAN
Struktur Data Linear
Searching Object dan Interpolation Search



NAMA : Johanes Yogtan Wicaksono Raharja
NIM : 215314105

Program Studi INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA

1. Diagram UML



2. Searching Object

a. Kelas Mahasiswa

i. Listing Program

```
package Modul10_SearchingObject;

public class Mahasiswa implements Comparable {
    private int nim;
    private String nama;

    public Mahasiswa(int nim, String nama) {
        this.nim = nim;
        this.nama = nama;
    }

    public Mahasiswa(int nim) {
        this.nim = nim;
    }

    public int getNim() {
        return nim;
    }

    public void setNim(int nim) {
        this.nim = nim;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String toString() {
        return "Data Mahasiswa {" + "NIM = " + nim + ", Nama = " + nama + '}';
    }

    public int compareTo(Object o) {
        if (((Mahasiswa) o).getNim() == nim) {
            return 0;
        } else if (((Mahasiswa) o).getNim() <= nim) {
            return 1;
        }
        return -1;
    }
}
```

ii. Penjelasan

Kelas mahasiswa merupakan kelas yang digunakan untuk menyimpan data nim dan nama. Kelas ini memiliki konstruktor dengan parameter yang berbeda yang digunakan untuk menyiapkan data pada kelas, selanjutnya metode set get pada nim dan nama untuk menyimpan dan menampilkan data nantinya, selanjutnya ada metode toString untuk menampilkan kalimat data variabel dan yang terakhir metode compareTo untuk membandingkan kemungkinan objek bilangan yang dipanggil dengan jika objek di kelas mahasiswa nimnya sama, maka akan mengembalikan 0, jika nimnya kurang maka akan mengembalikan 1, dan jika tidak keduanya akan mengembalikan -1.

b. Sequential Search

i. Listing Program

```
public static int sequentialSearchObject(Object[] mhs, Object key) {  
    for (int counter = 0; counter < mhs.length; counter++) {  
        if (((Comparable) mhs[counter]).compareTo(key) == 0) {  
            return counter;  
        }  
    }  
  
    return -1;  
}
```

ii. Penjelasan

Metode ini diikuti dengan parameter ibarat sebuah Object array mhs dan objek key, dilanjutkan dengan perulangan yang didalamnya ada percabangan yang dimana jika indeks array objek mhs sama dengan objek key sama dengan 0 maka akan mengembalikan indeks saat itu juga. Jika tidak ada yang dicari akan mengembalikan nilai -1

c. Binary Search

i. Listing Program

```
public static int BinarySearch(Object[] x, Object y) {  
    int Awal = 0;  
    int Akhir = x.length;  
  
    while (Awal <= Akhir) {  
        int tengah = (Awal + Akhir) / 2;  
        if (((Comparable) x[tengah]).compareTo(y) == 0) {  
            return tengah;  
        } else {  
            if (((Comparable) x[tengah]).compareTo(y) == 1) {  
                Akhir = (tengah - 1);  
            } else {  
                Awal = (tengah + 1);  
            }  
        }  
    }  
  
    return -1;  
}
```

ii. Penjelasan

Metode ini diikuti dengan parameter ibarat sebuah Object array mhs dan objek key, selanjutnya deklarasi atribut indek awal 0 dan akhir yang diisi panjang array objek x. Dilanjutkan sebuah perulangan dari awal ke akhir yang didalamnya ada ada pencarian nilai tengah dari atribut awal dan akhir, dilanjutkan percabangan yang dimana jika indeks array objek x sama dengan objek y sama dengan 0 maka akan mengembalikan indeks tengah dan jika tidak akan melakukan percabangan lagi didalam percabangan yang dimana jika indeks array objek mhs sama dengan objek key sama dengan 1 maka akan mencari nilai akhir dan jika tidak akan mencari nilai awal. Jika tidak ada yang dicari akan mengembalikan nilai -1.

d. Kelas Main

i. Listing Program

```
package Modul10_SearchingObject;
import java.util.Arrays;
import java.util.Scanner;

public class MainSequentialBinarySearch {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        Mahasiswa mhs[] = new Mahasiswa[3];
        mhs[0] = new Mahasiswa(215314105, "Yogtan");
        mhs[1] = new Mahasiswa(777777777, "Ronaldo");
        mhs[2] = new Mahasiswa(101010101, "Messi");

        System.out.println("Data Mahasiswa : ");
        Arrays.sort(mhs);
        Larik.cetak(mhs);

        System.out.println("\nSequential Search");
        System.out.print("Pencarian data : ");
        int data = sc.nextInt();
        Mahasiswa search = new Mahasiswa(data);

        int hasil = Larik.sequentialSearchObject(mhs, search);
        if (hasil == -1) {
            System.out.println(search.getNim() + " Tidak Terdapat Data Mahasiswa");
        } else {
            System.out.println("Data Mahasiswa Yang Dicari Pada Indeks = " + hasil);
        }

        System.out.println("\nBinary Search");
        System.out.print("Pencarian data : ");
        int data1 = sc.nextInt();
        Mahasiswa search1 = new Mahasiswa(data1);

        hasil = Larik.BinarySearch(mhs, search1);
        if (hasil == -1) {
            System.out.println(search1.getNim() + " Tidak Terdapat Data Mahasiswa");
        } else {
            System.out.println("Data Mahasiswa Yang Dicari Pada Indeks = " + hasil);
        }
    }
}
```

ii. Penjelasan

Pada kelas ini saya membuat sebuah objek array yang berisi 3 dan diisi sesuai dengan konstruktor pada kelas sebelumnya, selanjutnya melakukan pengurutan angka dengan memanggil pengurutan java dan memanggil metode cetak di kelas larik untuk menampilkan data yang telah diisi. Dilanjutkan dengan menginput pencarian data yang mana disimpan ke sebuah objek searching, selanjutnya memanggil metode sequential di kelas larik untuk melakukan pencarian dengan nilai array mhs dan objek search sesuai parameter yang dibuat di kelas larik dan hasilnya akan disimpan di atribut hasil untuk melihat pengembalian nilainya, dilanjutkan dengan percabangan dari hasil pengembalian nilai untuk melihat ada atau tidak datanya. Selanjutnya program sama penjelasannya seperti yang sudah dijelaskan yang membedakan hanya menggunakan metode binary saja.

3. Output

a. Hasil Output

```
run:
Data Mahasiswa :
Data Mahasiswa {NIM = 101010101, Nama = Messi}
Data Mahasiswa {NIM = 215314105, Nama = Yogtan}
Data Mahasiswa {NIM = 777777777, Nama = Ronaldo}

Sequential Search
Pencarian data : 101010101
Data Mahasiswa Yang Dicari Pada Indeks = 0

Binary Search
Pencarian data : 777777777
Data Mahasiswa Yang Dicari Pada Indeks = 2
BUILD SUCCESSFUL (total time: 19 seconds)
```

b. Penjelasan program keseluruhan

Pembuatan searching objek ini digunakan untuk mencari sebuah data menggunakan objek yang telah di deklarasikan. Untuk pembuatan searching objek ini pertama-tama saya membuat sebuah kelas mahasiswa untuk menyimpan sebuah datanya yang juga menggunakan sebuah implements interface comparable dari java untuk keperluan pencarian membandingkan objek yang dicari. Setelah itu kita dapat membuat kelas Larik yang berisi metode pencarian yang kita inginkan misalnya disini saya membuat metode pencarian sequential dan binary. Setelah itu kita membuat kelas mainnya dengan pembuatan array objek yang telah diisi sesuai kemauan, dilanjutkan dengan user menginputkan datanya yang dimana data tersebut dimasukkan ke sebuah objek, yang dilanjutkan dengan pemanggilan pencariannya objek di kelas larik untuk memasukkan objek array yang telah kita buat dan objek yang dicari, sehingga kita dapat melakukan pencarian data.