

# LAPORAN PRAKTIKUM

## STRUKTUR DATA LINIER

### MODUL III

Dosen Pengampu

JB. Budi Darmawan S.T., M.Sc.



DISUSUN OLEH :

AGUSTINUS KEVIN YUDIPRATAMA

235314029

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA**

**2024**

No 1

### A. TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami algoritma pencarian
- Mahasiswa mampu membuat program pencarian dengan tipe data sederhana (primitive)

### B. ALGORITMA

- Menginput data
- Cetak data
- Inisialisasi Variabel
- Pencarian sequential
  - Untuk setiap elemen dalam larik
  - Jika elemennya sama dengan elemen yang dicari
  - Set nilai hasil jadi indeks
  - Hentikan pencarian
- Output data

### C. DIAGRAM UML

Larik
+ cetak(int[ ] ) : static void

### D. SOURCE CODE

Class utamanya

```
package vscode.Modul3;

import vscode.Modul1.Larik;

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

        int data[] = {5, 8, 26, 15, 11, 31} ;

        System.out.println(x:"Data: ");
        Larik.cetak(data);

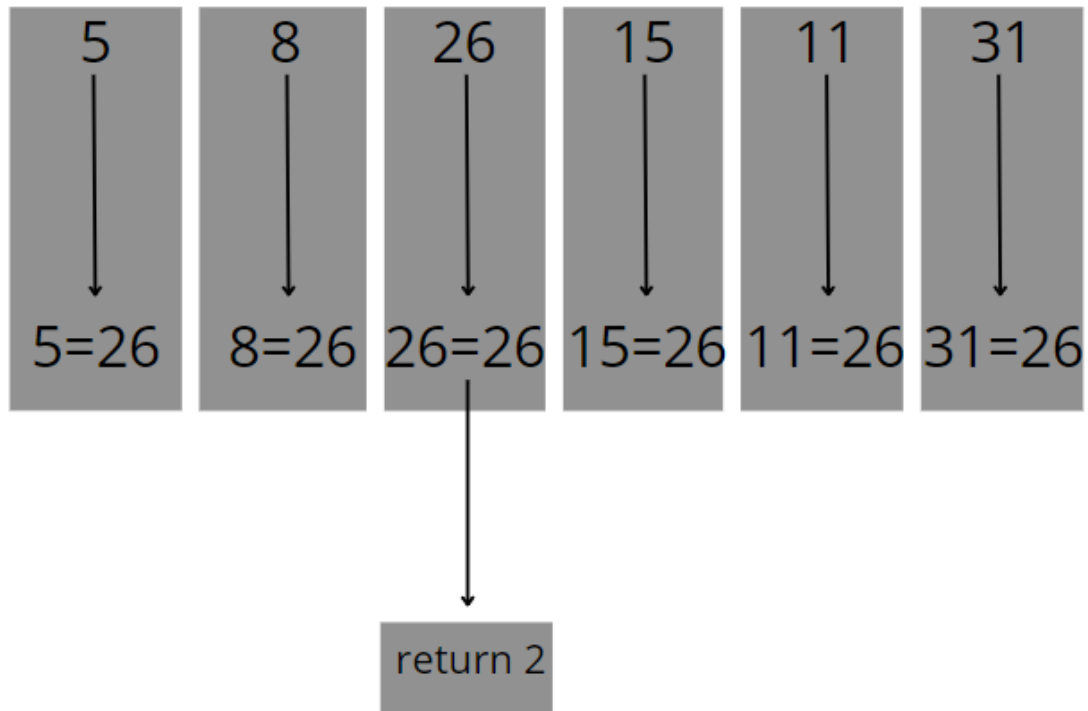
        System.out.println(x:"");

        int key = 26;
        int hasil = Larik.sequentialSearch(data, key);
        if (hasil >=0) {
            System.out.println("Data: " + key + " ada di index ke: " + hasil);
        }else{
            System.out.println("Data: " + key + " Tidak ditemukan");
        }
    }
}
```

### E. OUTPUT

```
Data:  
5 8 26 15 11 31  
Data: 26 ada di index ke: 2
```

### F. ANALISIS



No 2

### A. TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami algoritma pencarian
- Mahasiswa mampu membuat program pencarian dengan tipe data sederhana (primitive)

### B. ALGORITMA

- Menginput data
- Cetak data
- Inisialisasi Variabel
- Pencarian Binary
  - Untuk setiap elemen dalam larik
  - Jika elemennya sama dengan elemen yang dicari
  - Set nilai hasil jadi indeks
  - Hentikan pencarian
- Output data

### C. DIAGRAM UML

BinarySearch
-Larik: int[ ]
+BinarySearch(Larik: int[ ], kunci: int ): int

### D. SOURCE CODE

```
package vscode.Modul3;

import java.util.Arrays;

import vscode.Modul1.Larik;

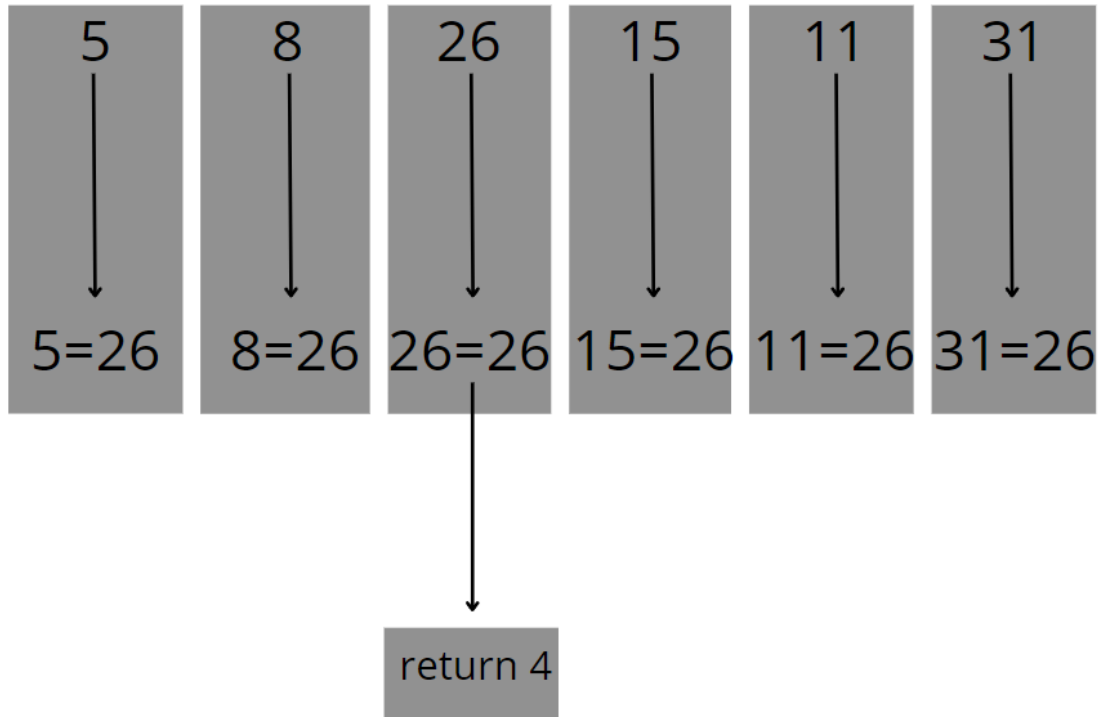
public class BinaryMain {
    Run | Debug
    public static void main(String[] args) {
        int data [] = {5, 8, 26, 15, 11, 31};
        Arrays.sort(data); // untuk mengurutkan

        int key = 26;
        int hasil = Larik.BinarySearch(data, key);
        if (hasil >= 0) {
            System.out.println("Data ditemukan di indeks ke- " + hasil );
        } else{
            System.out.println(x:"data tidak ditemukan");
        }
        //System.out.println();
        // System.out.println(Larik.BinarySearch(data, 11));
    }
}
```

**E. OUTPUT**

Data ditemukan di indeks ke- 4

**F. ANALISIS**



No 3

#### A. TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami algoritma pencarian
- Mahasiswa mampu membuat program pencarian dengan tipe data sederhana (primitive)

#### B. ALGORITMA

- Menginput data
- Cetak data
- Inisialisasi Variabel
- Pencarian interpolasi
- Output data

#### C. DIAGRAM UML

interpolasiSearch
-data:int[ ] -key: int -low: int -high: int
+interpolasiSearch(data: int[ ], key: int): int

#### D. SOURCE CODE

```
import java.util.Arrays;

import vscode.Modul1.Larik;

public class interpolasiMain {
    Run | Debug
    public static void main(String[] args) {
        int data [] = {5, 8, 26, 15, 11, 31};
        Arrays.sort(data);

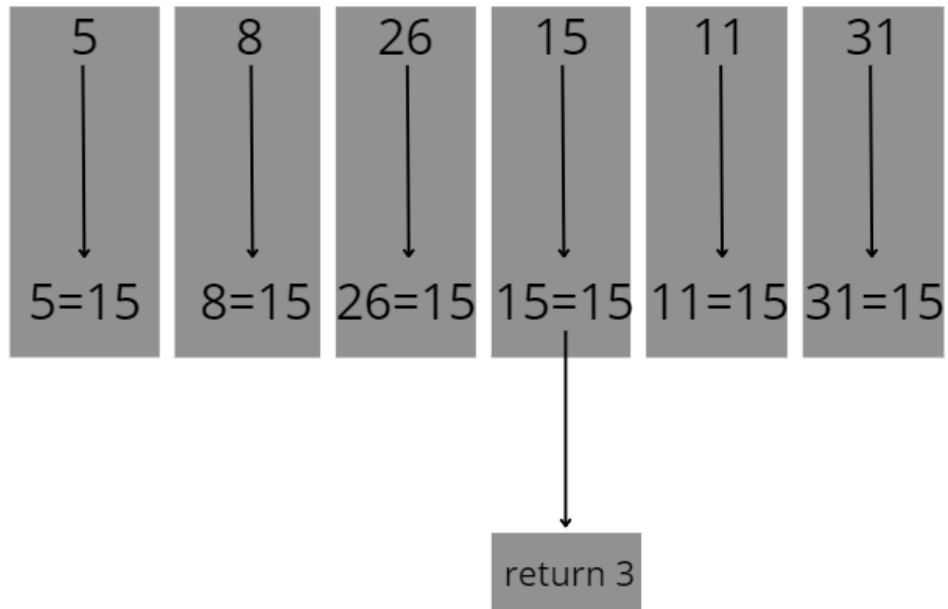
        int key = 15;
        int hasil = Larik.interpolasiSearch(data, key);

        if (hasil >= 0) {
            System.out.println("Data ditemukan pada indeks: " + hasil);
        }else{
            System.out.println(x:"Data tidak ditemukan");
        }
    }
}
```

### E. OUTPUT

Data ditemukan pada indeks: 3

### F. ANALISIS



## UNTUK SELURUH CLASS METODE DI LARIK

```
public class Larik {
    double array;

    public static void cetak(int[]x){
        for(int i = 0; i < x.length; i++){
            System.out.print(x[i]+" ");
        }
    }
}

public static void cetak2(int[]array){

    for (int counter = 0; counter < array.length;counter++){
        System.out.printf(format:"%5d%8d\n", counter, array[counter]);
    }

}

public static void cetak3(int[]i){
    for (int value : i)
        System.out.printf(format:" %d", value);
}

public static void modifyArray(int[]array2){
    for (int counter = 0; counter < array2.length;counter++)
        array2[counter]*=2;
}

public static void modifyElement(int element)
{
    element *=2;
    System.out.printf(
        format:"value of element in modifyElement: %d\n", element
    );
}

public static int sequentialSearch(int[] Larik, int kunci){
    for(int counter = 0;counter <Larik.length;counter++) {
        if (kunci ==Larik[counter]){
            return counter;
        }
    }

    return -1;
}

public static int BinarySearch(int [] Larik, int kunci){
    int indexAwal = 0;
    int indexAkhir = Larik.length;
    int indexTengah;

    while (indexAwal <= indexAkhir) {
        indexTengah = (indexAwal + indexAkhir)/2;
        if (Larik[indexTengah] == kunci) {
            return indexTengah;
        }
        else {
            if (Larik[indexTengah]> kunci) {
                indexAkhir = indexTengah -1;
            }
            else{
                indexAwal = indexTengah +1;
            }
        }
    }

    return -1;
}

public static int interpolasiSearch(int [] data, int key){
    int low = 0;
    int high = data.length -1;
    while (low <= high && key >= data[low] && key <= data[high]) {
        int mid = low + ((key - data[low])*(high - low))/ (data[high]- data[low]);

        if (data[mid]==key) {
            return mid;
        }
        if (data[mid] > key) {
            high = mid -1;
        }
        else{
            low = mid +1;
        }
    }
}
```



