

UTS ASD

Oleh:

FABIO CHRISTIANO MALISAN NIM:1941720202



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG MARET
2020**

Seorang pustakawan hendak mengurutkan buku di dalam almari secara berurutan dari terpendek-tertipis hingga tertinggi-tertebal secara vertikal (berdiri). Buat simulasi pengurutan dan pencarian yang dilakukan oleh pustakawan tersebut dengan mengacu data pada tabel, aturan pengurutan berdasarkan metode program, dan class diagram berikut!

Objek: Ukuran buku menggunakan merge sort untuk pengurutan dan sequential search untuk pencarian

Catatan: Ukuran buku harus melalui konversi tinggi buku yang dimaksud dalam satuan ukuran, contoh A4: 21,0 x 29,7 cm sehingga nilai tinggi buku yang terpakai adalah 29,7.

| NO | NIM | NAMA | Soal |
|----|-----|------|------|
|----|-----|------|------|

Judul Buku Jumlah Halaman

Ukuran Buku

Introduction to Algorithms

1292 B5

Essentials of Computer Architecture

400 A4

A Life out of Sequence — A Data Driven History of Bioinformatics

294 B3

Engineering a Compiler 2nd Ed

824 B5

Buku

judul: String jumlahHalaman: int tinggiBuku: double

Buku(j: String, jml: int, tg: double)

tampil(): void bubbleSort(): void binarySearch(): void

```

*
* @author ACER
*/
class DaftarBuku {
    buku listBuku[] = new buku[5];
    public int[] data;
    int idx;
    int lenght;
    int length;

    void tambah(buku b) {
        if (idx < listBuku.length) {
            listBuku[idx] = b;
            idx++;
        } else {
            System.out.print("Data sudah penuh!!");
        }
    }

    public void tampil() {
        for (buku b : listBuku) {
            b.tampil();
            System.out.println("=====");
        }
    }

    public void bubbleSort() {
        for (int i = 0; i < listBuku.length - 1; i++) {
            for (int j = 1; j < listBuku.length - i; j++) {
                if (listBuku[j].tinggiBuku > listBuku[j - 1].tinggiBuku) {
                    buku tmp = listBuku[j];
                    listBuku[j] = listBuku[j - 1];
                    listBuku[j - 1] = tmp;
                }
            }
        }
    }
}

```

```

    public void bubbleSort() {
        for (int i = 0; i < listBuku.length - 1; i++) {
            for (int j = 1; j < listBuku.length - i; j++) {
                if (listBuku[j].tinggiBuku > listBuku[j - 1].tinggiBuku) {
                    buku tmp = listBuku[j];
                    listBuku[j] = listBuku[j - 1];
                    listBuku[j - 1] = tmp;
                }
            }
        }
    }

    public int binarySearch(int cari, int left, int right) {
        int mid;
        if (right >= left) {
            mid = (left + right) / 2;
            if (cari == data[mid]) {
                return mid;
            } else if (data[mid] > cari) {
                return binarySearch(cari, left, mid - 1);
            } else {
                return binarySearch(cari, mid + 1, right);
            }
        }
        return 0;
    }

    void tampil() {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package uts.asd;

/**
 *
 * @author ACER
 */
class buku {
    public String judul;
    public int jumlahHalaman;
    public double tinggiBuku;
    private String[] listBuku;

    public buku(String j, int jml, double tg) {
        this.judul = j;
        this.jumlahHalaman = jml;
        this.tinggiBuku = tg;
    }

    buku(DaftarBuku data, int i) {
    }

    public void tampil() {
        System.out.println("Judul = "+judul);
        System.out.println("Jumlah Halaman = "+jumlahHalaman);
        System.out.println("Tinggi Buku = "+tinggiBuku);

        for (int i = 0; i < jumlahHalaman; i++) {
            System.out.print(listBuku[i] + " ");
        }
        System.out.println();
    }

    public void TampilPosisi(int x, int pos) {
        if (pos != -1) {
            System.out.println("data : " + x + " tidak ditemukan");
        }
        else{
            System.out.println("data " + x + " tidak ditemukan");
        }
    }

    public int binarySearch(int cari) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated method
    }
}

```

```

import java.util.Scanner;

/**
 *
 * @author ACER
 */
public class BukuMain {
    public static void main(String[] args){
        Scanner s = new Scanner (System.in);
        Scanner s1 = new Scanner (System.in);
        DaftarBuku data = new DaftarBuku();
        int jumBuku = 4;
        buku pencarian = new buku(data, 8);
        System.out.println("isi Array : ");
        pencarian.tampil();

        for (int i = 0; i < jumBuku; i++) {
            System.out.print("Judul = ");
            String judul = s1.nextLine();
            System.out.print("Jumlah halaman = ");
            int jmlHal = s.nextInt();
            System.out.print("Tinggi Buku = ");
            int tgBuku = s.nextInt();

            buku m = new buku(judul, jmlHal, tgBuku);
            data.tambah(m);

        }

        System.out.println("Data buku sebelum sorting");
        data.tampil();
        System.out.println("Data buku setelah diurutkan : ");
        data.tampil();
        System.out.print("Judul yang ingin dicari : ");
        int cari = s1.nextInt();
        int posisi = pencarian.binarySearch(cari);

        pencarian.TampilPosisi(cari, posisi);
        System.out.println("=====");
        System.out.println("menggunakan binary search");
        posisi = pencarian.binarySearch(cari, 0, data.length - 1);
        pencarian.TampilPosisi(cari, posisi);
    }
}

```

OUTPUT

```
output ×
UTS ASD (run) × UTS ASD (run) #2 ×

run:
isi Array :
Judul = null
Jumlah Halaman = 0
Tinggi Buku = 0.0

Judul = MM
Jumlah halaman = 12
Tinggi Buku = 12
Judul = 12
```