

**LAPORAN TUGAS PRAKTIKUM  
SEARCHING**



**Oleh:**

**FAWWAZ ALIFIO FARSA**

**NIM. 2341720128**

**TI-1E / 10**

**D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

## Praktikum Percobaan 1

### Verifikasi Hasil Percobaan

```
-----  
Masukkan data Buku secara urut dari KodeBuku terkecil:  
-----
```

```
Kode buku      : 111  
Judul buku     : Algoritma  
Tahun terbit   : 2019  
Pengarang     : Wahyuni  
Stock         : 5  
-----
```

```
Kode buku      : 123  
Judul buku     : Big Data  
Tahun terbit   : 2020  
Pengarang     : Susilo  
Stock         : 3  
-----
```

```
Kode buku      : 125  
Judul buku     : Desain UI  
Tahun terbit   : 2021  
Pengarang     : Supriadi  
Stock         : 3  
-----
```

```
Kode buku      : 126  
Judul buku     : Web Programming  
Tahun terbit   : 2022  
Pengarang     : Pustaka Adi  
Stock         : 2  
-----
```

```
Kode buku      : 127  
Judul buku     : Etika Mahasiswa  
Tahun terbit   : 2023  
Pengarang     : Darmawan Adi  
Stock         : 2  
-----
```

```
-----  
Pencarian Data:
```

```
Masukkan kode buku yang dicari:
```

```
Kode Buku: 111
```

```
Menggunakan sequential search
```

```
Data 111 ditemukan pada indeks 0
```

```
Kode buku      : 111  
Judul buku     : Algoritma  
Tahun terbit   : 2019  
Pengarang     : Wahyuni  
Stock         : 5
```

### Jawaban Pertanyaan Praktikum Percobaan 1

1. Fungsi **break** pada method **findSeqSearch** digunakan untuk menghentikan iterasi pada **for** loop ketika menemukan nilai yang sama dengan nilai **cari**.
2. Jika kode buku yang dimasukkan tidak terurut dari kecil ke besar, program masih akan tetap bisa berjalan dengan baik.

```

=====
Kode buku: 123
Judul buku: Big Data
Tahun terbit: 2020
Pengarang: Susilo
Stock: 3
=====
Kode buku: 111
Judul buku: Algoritma
Tahun terbit: 2019
Pengarang: Wahyuni
Stock: 5
=====
Kode buku: 125
Judul buku: Desain UI
Tahun terbit: 2021
Pengarang: Supriadi
Stock: 3
=====
Pencarian Data:
Masukkan kode buku yang dicari:
Kode Buku: 125
Menggunakan sequential search
Data 125 ditemukan pada indeks 2
Kode buku      : 125
Judul buku     : Desain UI
Tahun terbit   : 2021
Pengarang      : Supriadi
Stock          : 3

```

Hal tersebut terjadi karena sequential search akan tetap mencari nilai yang sesuai dengan nilai **cari** dari awal array hingga akhir array.

3. Membuat method baru dengan nama **findBuku** menggunakan konsep sequential search dengan tipe methodnya adalah objek **Buku10**.

```

public Buku10 findBuku(int cari) {
    Buku10 bukuDitemukan = null;
    for (int i = 0; i < listBk.length; i++) {
        if (listBk[i].kodeBuku == cari) {
            bukuDitemukan = listBk[i];
            break;
        }
    }
    return bukuDitemukan;
}

```

```

Buku10 dataBuku = data.findBuku(cari);
dataBuku.tampilDataBuku();

```

## Praktikum Percobaan 2

### Verifikasi Hasil Percobaan

```
Kode Buku: 126
Menggunakan sequential search
=====
Kode buku: 126
Judul buku: Web Programming
Tahun terbit: 2022
Pengarang: Pustaka Adi
Stock: 2
=====
Menggunakan binary search
Data 126 ditemukan pada indeks 3
Kode buku      : 126
Judul buku     : Web Programming
Tahun terbit   : 2022
Pengarang      : Pustaka Adi
Stock          : 2
```

### Jawaban Pertanyaan Praktikum Percobaan 2

1. Proses divide dijalankan pada kode **mid = (left + right) / 2**.
2. Proses conquer dijalankan pada kode **return findBinarySearch(cari, left, mid-1);** dan **return findBinarySearch(cari, mid+1, right);**.
3. Jika kode buku yang dimasukkan tidak urut, maka program tidak akan berjalan dengan baik karena algoritma binary search sangat bergantung pada sifat data terurut untuk membagi rentang pencarian menjadi dua bagian dan mengurangi ukuran rentang tersebut pada setiap iterasi.

```
Menggunakan binary search
Data 126 ditemukan pada indeks -1
Kode buku      : 126
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -1 out of bounds for length 5
    at Jobsheet6.TestSearching.PencarianBuku10.tampilData(PencarianBuku10.java:44)
    at Jobsheet6.TestSearching.BukuMain10.main(BukuMain10.java:53)
```

4. Mengubah kode program agar binary search dapat melakukan pencarian dengan urutan kode buku terbesar ke terkecil.

```
public int findBinarySearch(int cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (left + right) / 2;
        if (cari == listBk[mid].kodeBuku) {
            return (mid);
        } else if (listBk[mid].kodeBuku > cari) {
            return findBinarySearch(cari, mid + 1, right);
        } else {
            return findBinarySearch(cari, left, mid - 1);
        }
    }
    return -1;
}
```

### Praktikum Percobaan 3

#### Verifikasi Hasil Percobaan

```
fiota@LAPTOP-NATDJTJ6 MINGW64 /c/Pio's/College/Semester 2/Algoritma dan Struktur Data (master)
$ /usr/bin/env C:\\Program\\Files\\Java\\jdk-20\\bin\\java.exe -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\fiota\\AppData\\Roaming\\Code\\User\\workspaceStorage\\af0b9ed7e0f2d784a54a7bc4ade81d55\\redhat.java\\jdt_ws\\Algoritma\\ dan\\ Struktur\\ Data_ead2f687\\bin Jobsheet6.MergeSortTest
.MergeSortMain10
Sorting dengan merge sort
Data awal
10 40 30 50 70 20 100 90
Setelah diurutkan
10 20 30 40 50 70 90 100
```

### Latihan Praktikum

1. Mengubah percobaan searching dengan ketentuan sebagai berikut:
  - Mengubah tipe data dari kodeBuku yang awalnya int menjadi String.

```
public class Buku10 {
    String kodeBuku;
    String judulBuku;
    int tahunTerbit;
    String pengarang;
    int stock;

    public Buku10(String kodeBuku, String judulBuku, int tahunTerbit, String pengarang, int stock) {
        this.kodeBuku = kodeBuku;
        this.judulBuku = judulBuku;
        this.tahunTerbit = tahunTerbit;
        this.pengarang = pengarang;
        this.stock = stock;
    }
}
```

- Menambahkan method untuk pencarian kodeBuku (bertipe data String) dengan menggunakan sequential search dan binary search.

```
public Buku10 findSeqSearchString(String cari) {
    Buku10 bukuDitemukan = null;
    for (int i = 0; i < listBk.length; i++) {
        if (listBk[i].kodeBuku.equalsIgnoreCase(cari)) {
            bukuDitemukan = listBk[i];
            break;
        }
    }
    return bukuDitemukan;
}

public Buku10 findBinarySearch(String cari, int left, int right) {
    Buku10 bukuDitemukan = null;
    int mid;
    if (right >= left) {
        mid = (left + right) / 2;
        if (cari.equalsIgnoreCase(listBk[mid].kodeBuku)) {
            bukuDitemukan = listBk[mid];
            return bukuDitemukan;
        } else if (Integer.parseInt(listBk[mid].kodeBuku) > Integer.parseInt(cari)) {
            return findBinarySearch(cari, mid + 1, right);
        } else {
            return findBinarySearch(cari, left, mid - 1);
        }
    }
    return bukuDitemukan;
}
```

2. Memodifikasi percobaan searching di atas dengan ketentuan berikut:

- Menambahkan method pencarian judul buku menggunakan sequential search dan binary search.

```
public Buku10 findSeqSearchByJudul(String cari) {
    bubbleSort();
    Buku10 bukuDitemukan = null;
    for (int i = 0; i < listBk.length; i++) {
        if (listBk[i].judulBuku.equalsIgnoreCase(cari)) {
            bukuDitemukan = listBk[i];
            break;
        }
    }
    return bukuDitemukan;
}

public Buku10 findBinarySearchByJudul(String cari, int left, int right) {
    bubbleSort();
    Buku10 bukuDitemukan = null;
    int mid;
    if (right >= left) {
        mid = (left + right) / 2;
        if (cari.equalsIgnoreCase(listBk[mid].judulBuku)) {
            bukuDitemukan = listBk[mid];
            return bukuDitemukan;
        } else if (Integer.parseInt(listBk[mid].judulBuku) > Integer.parseInt(cari)) {
            return findBinarySearch(cari, mid + 1, right);
        } else {
            return findBinarySearch(cari, left, mid - 1);
        }
    }
    return bukuDitemukan;
}
```

- Menambahkan method untuk mendeteksi hasil pencarian duplikasi buku.

```
public void findAndWarnDuplicateTitles(String cari) {
    int count = 0;
    int firstIndex = -1;

    for (int i = 0; i < listBk.length; i++) {
        if (listBk[i].judulBuku.equalsIgnoreCase(cari)) {
            count++;
            if (count == 1) {
                firstIndex = i;
            }
        }
    }

    if (count > 1) {
        System.out.println("Peringatan: Lebih dari satu judul buku dengan judul '" + cari + "' ditemukan!");
        System.out.println("Judul buku ditemukan pada indeks:");
        System.out.println(firstIndex);
    }
}
```