



**Nama**

Raudhil Firdaus Naufal

**NIM**

2341720164

**Kelas**

1E

**Jurusan**

Teknologi Informasi

**Prodi**

Teknik Informatika

## Jobsheet 6: Searching

### Percobaan 1

#### Hasil Percobaan:

```
Data keseluruhan Buku:
=====
Kode buku: 111
Judulu buku: Algoritma
Tahun terbit: 2019
Pengarang: Wahyuni
Stock: 5
=====
Kode buku: 123
Judulu buku: Big Data
Tahun terbit: 2020
Pengarang: Susilo
Stock: 3
=====
Kode buku: 125
Judulu buku: Desain UI
Tahun terbit: 2021
Pengarang: Supriadi
Stock: 3
=====
Kode buku: 126
Judulu buku: Web Programming
Tahun terbit: 2022
Pengarang: Pustaka Adi
Stock: 2
=====
Kode buku: 127
Judulu buku: Etika Mahasiswa
Tahun terbit: 2023
Pengarang: Dermawan adi
Stock: 2
-----
-----
```

#### Pencarian Data:

Masukkan kode buku yang dicari:

Kode Buku: 111

Menggunakan sequential Search

Data: 111 ditemukan pada indeks 0

#### Pencarian Data:

Masukkan kode buku yang dicari:

Kode Buku: 124

Menggunakan sequential Search

Data 124 tidak ditemukan

Data 124 tidak ditemukan

### Pertanyaan:

1. Jelaskan fungsi break yang ada pada method FindSeqSearch!  
Fungsinya adalah ketika kita sudah menemukan kode buku yang kita inginkan, maka perulangan akan terhenti sehingga tidak harus mencari kode buku di atasnya.
2. Jika Data Kode Buku yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Tunjukkan hasil screenshoot untuk bukti dengan kode Buku yang acak. Jelaskan Mengapa hal tersebut bisa terjadi?

Ya program masih dapat berjalan.

```
Data keseluruhan Buku:
=====
Kode buku: 123
Judulu buku: Big Data
Tahun terbit: 2020
Pengarang: Susilo
Stock: 3
=====
Kode buku: 111
Judulu buku: Algoritma
Tahun terbit: 2019
Pengarang: Wahyuni
Stock: 5
=====
Kode buku: 125
Judulu buku: Desain UI
Tahun terbit: 2021
Pengarang: Supriadi
Stock: 3
=====
Kode buku: 127
Judulu buku: Etika Mahasiswa
Tahun terbit: 2023
Pengarang: Darmawan Adi
Stock: 2
=====
Kode buku: 126
Judulu buku: Web Programming
Tahun terbit: 2022
Pengarang: Pustaka Adi'
Stock: 2
```

```
Pencarian Data:
Masukkan kode buku yang dicari:
Kode Buku: 111
Menggunakan sequential Search
Data: 111 ditemukan pada indeks 1
```

Hasilnya benar, mengapa? Karena walaupun kita memasukkan kode secara acak, kode buku akan tetap dicari tanpa memedulikan rapi atau tidaknya, asalkan kode buku yang dicari tersedia.

3. Buat method baru dengan nama FindBuku menggunakan konsep sequential search dengan tipe method dari FindBuku adalah BukuNoAbsen. Sehingga Anda bisa memanggil method tersebut pada class BukuMain seperti gambar berikut :

```
public buku22 findBuku(int x){  
    for (int i = 0; i < listBk.length; i++) {  
        if (listBk[i].kodeBuku == x) {  
            return listBk[i];  
        }  
    }  
    return null;  
}
```

```
buku22 dataBuku = data.findBuku(cari);  
if (dataBuku == null) {  
    System.out.println("Kode Buku Tidak Tersedia");  
} else{  
    dataBuku.tampilDataBuku();  
}
```

## Percobaan 2:

### Hasil Percobaan:

```
Pencarian Data:  
Masukkan kode buku yang dicari:  
Kode Buku: 126  
Menggunakan sequential Search  
Data: 126 ditemukan pada indeks 3  
Kode Buku      : 126  
Judul   : Web Programming  
Tahun Terbit  : 2022  
Pengarang    : Pustaka Adi  
Stock   : 2  
=====
```

```
=====
```

```
Menggunakan binary Search  
Data: 126 ditemukan pada indeks 3  
Kode Buku      : 126  
Judul   : Web Programming  
Tahun Terbit  : 2022  
Pengarang    : Pustaka Adi  
Stock   : 2
```

**Pertanyaan:**

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

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

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

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

3. Jika data Kode Buku yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian! Tunjukkan hasil screenshoot untuk bukti dengan kode Buku yang acak. Jelaskan Mengapa hal tersebut bisa terjadi?

Tidak program tidak bisa berjalan karena pada Binary Search, dilakukan proses membandingkan, oleh karena itu harus berurutan dari yang terbesar menjadi yang terkecil ataupun sebaliknya.

```
Kode Buku      : 222
Judul buku     : 222
Tahun Terbit   : 222
Pengarang      : 222
Stock   : 222
-----
Kode Buku      : 444
Judul buku     : 444
Tahun Terbit   : 444
Pengarang      : 444
Stock   : 444
-----
Kode Buku      : 333
Judul buku     : 333
Tahun Terbit   : 333
Pengarang      : 333
Stock   : 333
-----
Kode Buku      : 111
Judul buku     : 111
Tahun Terbit   : 111
Pengarang      : 111
Stock   : 111
```

[illegible]

4. Jika Kode Buku yang dimasukkan dari Kode Buku 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 search agar hasilnya sesuai!

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

Data keseluruhan Buku:

```
=====
Kode buku: 20215
Judulu buku: 1
Tahun terbit: 1
Pengarang: 1
Stock: 1
=====
Kode buku: 20214
Judulu buku: 2
Tahun terbit: 2
Pengarang: 2
Stock: 2
=====
Kode buku: 20212
Judulu buku: 3
Tahun terbit: 3
Pengarang: 3
Stock: 3
=====
Kode buku: 20211
Judulu buku: 4
Tahun terbit: 4
Pengarang: 4
Stock: 4
=====
Kode buku: 20210
Judulu buku: 5
Tahun terbit: 5
Pengarang: 5
Stock: 5
```

Pencarian Data:

```
Masukkan kode buku yang dicari:
Kode Buku: 20210
Menggunakan sequential Search
Data: 20210 ditemukan pada indeks 4
Kode Buku      : 20210
Judul   : 5
Tahun Terbit : 5
Pengarang : 5
Stock   : 5
=====

Menggunakan binary Search
Data: 20210 ditemukan pada indeks 4
Kode Buku      : 20210
Judul   : 5
Tahun Terbit : 5
Pengarang : 5
Stock   : 5
```

### Percobaan 3:

#### Hasil Percobaan:

```
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. Modifikasi percobaan searching diatas dengan ketentuan berikut ini
- Ubah tipe data dari kode Buku yang awalnya int menjadi String.

```
System.out.print(s:"Kode Buku \t: ");
String kodeBuku = sc.next();
```

- Tambahkan method untuk pencarian kode Buku (bertipe data String) dengan menggunakan sequential search dan binary search.

Sequential Search:

```
public int findJudulSeq(String cari){
    int posisi = -1;
    for (int i = 0; i < listBk.length; i++) {
        if (listBk[i].judulBuku.equalsIgnoreCase(cari)) {
            posisi = i;
        }
    }
}
```

Binary Search:

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

2. Modifikasi percobaan searching diatas dengan ketentuan berikut ini
- Tambahkan method pencarian judul buku menggunakan sequential search dan binary search. Sebelum dilakukan searching dengan binary search data harus dilakukan pengurutan dengan menggunakan algoritma Sorting (bebas pilih algoritma sorting apapun)! Sehingga ketika input data acak, maka algoritma searching akan tetap berjalan

```
public void sort(){
    buku22 tmp = new buku22();
    for (int i = 0; i < listBk.length-1; i++) {
        for (int j = 1; j < listBk.length-i; j++) {
            if (Integer.parseInt(listBk[j].kodeBuku) > Integer.parseInt(listBk[j-1].
                kodeBuku)) {
                tmp = listBk[j];
                listBk[j] = listBk[j-1];
                listBk[j-1] = tmp;
            }
        }
    }
}
```

- Buat aturan untuk mendeteksi hasil pencarian judul buku yang lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

```
if (hitung == 1) {
    return posisi;
} else {
    System.out.println(x:"Input buku dengan Kode yang sama Lebih dari 1!");
    return -2;
}
```

```
if (mid >= 0 && mid < listBk.length - 1) {
    if (listBk[mid].judulBuku.equals(listBk[mid + 1].judulBuku)) {
        System.out.println(x:"Input buku dengan Kode yang sama Lebih dari
            1!");
        return -2;
    }
}
if (mid > 0 && mid <= listBk.length - 1) {
    if (listBk[mid].judulBuku.equals(listBk[mid - 1].judulBuku)) {
        System.out.println(x:"Input buku dengan Kode yang sama Lebih dari
            1!");
        return -2;
    }
}
```



```
Pencarian Data:
Masukkan kode buku yang dicari:
Kode Buku: 1
Menggunakan sequential Search
Input buku dengan Kode yang sama Lebih dari 1!
Data duplikat
=====

=====
Menggunakan binary Search
Input buku dengan Kode yang sama Lebih dari 1!
Data duplikat
```