

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN  
STRUKTUR DATA**

**JOBSHEET 6**



**MUHAMMAD AMMAR HAFIZH**

**(2341720074)**

**D-IV TEKNIK INFORMATIKA – 1E**

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

## Pertanyaan Percobaan 1

1. Jelaskan fungsi break yang ada pada method FindSeqSearch!

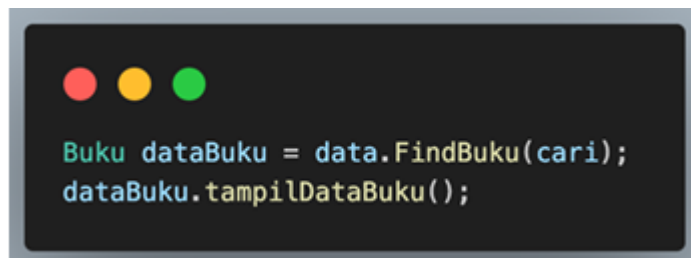
- Keluar dari loop setelah menemukan data yang cocok dan setelah itu mengembalikan nilai posisi terbaru.

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?

- Program tetap berjalan dengan benar karena menggunakan metode sequential search yaitu mengecek satu per-satu elemen yang ingin dicari dan memungkinkan tidak ada elemen yang terlewat.

```
Masukan Data Buku Secara Urut dari KodeBuku Terkecil :
-----
Kode Buku      : 127
Judul Buku     : Etika Mahasiswa
Tahun Terbit   : 2023
Pengarang      : Darmawan Adi
Stok           : 2
-----
Kode Buku      : 123
Judul Buku     : Big Data
Tahun Terbit   : 2020
Pengarang      : Susilo
Stok           : 3
-----
Kode Buku      : 126
Judul Buku     : Web Programming
Tahun Terbit   : 2022
Pengarang      : Pustaka Adi
Stok           : 2
-----
Kode Buku      : 111
Judul Buku     : Algoritma
Tahun Terbit   : 2019
Pengarang      : Wahyuni
Stok           : 5
-----
Kode Buku      : 125
Judul Buku     : Desain UI
Tahun Terbit   : 2021
Pengarang      : Supriadi
Stok           : 3
-----
Pencarian Data :
Masukan Kode Buku yang ingin dicari :
Kode Buku      : 111
Menggunakan Sequential Search
Data : 111 ditemukan pada indeks ke-3
```

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 :



- Code

```
public Buku16 findBuku(int cari){  
    for (int i = 0; i < listBk.length; i++) {  
        if (listBk[i].kodeBuku == cari) {  
            return listBk[i];  
        }  
    }  
    return null;  
}
```

- Hasil

```
Pencarian Data :  
Masukan Kode Buku yang ingin dicari :  
Kode Buku : 125  
Menggunakan Sequential Search  
Data : 125 ditemukan pada indeks ke-2  
=====  
Kode Buku : 125  
Judul : Desain UI  
Pengarang : Supriadi  
Tahun Terbit : 2021  
Stok : 3
```

## Pertanyaan Percobaan 2

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

- ```
mid = (left + 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 - 1);  
} else {  
    return FindBinarySearch(cari, mid + 1, right);  
}
```

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?

- Program dapat berjalan tetapi tidak menghasilkan nilai yang benar atau tidak akurat, Dalam searching dengan metode binary search diharuskan data dalam keadaan terurut descending maupun ascending.  
Berikut adalah contoh jika data tidak urut dan dilakukan pencarian data menggunakan metode binary search.

```
Pencarian Data :  
Masukan Kode Buku yang ingin dicari :  
Kode Buku : 4  
=====  
Menggunakan Sequential Search  
Data : 4 ditemukan pada indeks ke-4  
=====  
Kode Buku : 4  
Judul : 4  
Pengarang : 4  
Tahun Terbit : 4  
Stok : 4  
=====  
Menggunakan Binary Search  
Data : 4 tidak ditemukan
```

4. Jika Kode Buku yang dimasukkan dari Kode Buku terbesar ke terkecil (misal : 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!

- Tidak Sesuai
- Before

```
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, left, mid);
        } else {
            return FindBinarySearch(cari, mid, right);
        }
    }
    return -1;
}
```

- After

```
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, left, mid - 1);
        } else {
            return FindBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}
```

## Latihan Praktikum 1

1. Modifikasi percobaan searching diatas dengan ketentuan berikut ini

- Ubah tipe data dari kode Buku yang awalnya int menjadi String Tambahkan method untuk pencarian kode Buku (bertipe data String) dengan menggunakan sequential search dan binary search.

- Hasil

```
Pengarang      : 2
Stok           : 2
-----
Kode Buku      : 3
Judul Buku     : 3
Tahun Terbit   : 3
Pengarang      : 3
Stok           : 3
-----
Kode Buku      : 4
Judul Buku     : 4
Tahun Terbit   : 4
Pengarang      : 4
Stok           : 4
-----
Kode Buku      : 5
Judul Buku     : 5
Tahun Terbit   : 5
Pengarang      : 5
Stok           : 5
-----
Pencarian Data :
Masukan Kode Buku yang ingin dicari :
Kode Buku : 2
=====
Menggunakan Sequential Search
Data : 2 ditemukan pada indeks ke-1
=====
Kode Buku : 2
Judul : 2
Pengarang : 2
Tahun Terbit : 2
Stok : 2
=====
Menggunakan Binary Search
Data : 2 ditemukan pada indeks ke-1
Kode Buku      : 2
Judul           : 2
Tahun Terbit    : 2
Pengarang       : 2
Stok            : 2
```

## Latihan Praktikum 2

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

- Hasil Program

```
Stok          : 5
-----
Kode Buku     : 2
Judul Buku    : Qiqi
Tahun Terbit  : 2
Pengarang     : 2
Stok          : 2
-----
Kode Buku     : 1
Judul Buku    : Rhisky
Tahun Terbit  : 1
Pengarang     : 1
Stok          : 1
-----
Kode Buku     : 4
Judul Buku    : Ammar
Tahun Terbit  : 4
Pengarang     : 4
Stok          : 4
-----
Kode Buku     : 3
Judul Buku    : Tomo
Tahun Terbit  : 3
Pengarang     : 3
Stok          : 3
-----
Pencarian Data :
Masukan Judul Buku yang ingin dicari :
Rhisky
=====
Menggunakan Sequential Search
Data : Rhisky ditemukan pada indeks ke-2
Data Tidak Ditemukan
=====
Menggunakan Binary Search
Data : Rhisky ditemukan pada indeks ke-0
Kode Buku     : Rhisky
Judul         : Rhisky
Tahun Terbit  : 1
Pengarang     : 1
Stok          : 1
```

- 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!

- Hasil Program

```
-----
Kode Buku      : 1
Judul Buku     : Qiqi
Tahun Terbit   : 1
Pengarang      : 1
Stok           : 1
-----
Kode Buku      : 3
Judul Buku     : Qiqi
Tahun Terbit   : 3
Pengarang      : 3
Stok           : 3
-----
Kode Buku      : 2
Judul Buku     : Ammar
Tahun Terbit   : 2
Pengarang      : 2
Stok           : 2
-----
Kode Buku      : 5
Judul Buku     : Saep
Tahun Terbit   : 5
Pengarang      : 5
Stok           : 5
-----
Kode Buku      : 4
Judul Buku     : Tomo
Tahun Terbit   : 4
Pengarang      : 4
Stok           : 4
-----
Pencarian Data :
Masukan Judul Buku yang ingin dicari :
Qiqi
=====
Menggunakan Sequential Search
Data : Qiqi Ditemukan Lebih Dari 1
=====
Menggunakan Binary Search
Data : Qiqi Ditemukan Lebih Dari 1
```

Link : <https://github.com/uhamhz/Semester-2/tree/main/Algoritma%20Dan%20Struktur%20Data/minggu7>