

JOBSHEET VI SEARCHING

Nama: Rizqi Bagus Andrean

Kelas: TI-1D

Absen: 25

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Menjelaskan mengenai algoritma Searching.
2. Membuat dan mendeklarasikan struktur algoritma Searching.
3. Menerapkan dan mengimplementasikan algoritma Searching.

6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

Perhatikan diagram class Buku di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Buku.

Buku
kodeBuku: int judul: String tahunTerbit: int pengarang: String stock : int
Buku(kodeBuku:int, judul: String, tahunTerbit: int, pengarang: String, stock:int) tampil(): void TampilDataBuku():void

Berdasarkan class diagram di atas, akan dibuat class Buku yang berfungsi untuk membuat objek buku yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga Method tampil() untuk menampilkan semua attribute yang ada.

PencarianBuku
listBuku: Buku[5] idx: int
Tambah(bk: Buku): void tampil(): void FindSeqSearch(int cari): int Tampilpoisisi(int x,int pos): void TampilData(int x,int pos) :void

Selanjutnya class diagram PencarianBuku merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array Buku, misalkan untuk menambahkan



objek Buku, menampilkan semua data Buku, untuk melakukan pencarian berdasarkan kode Buku menggunakan algoritma Sequential Search, menampilkan posisi dari data yang dicari, serta menampilkan data Buku yang dicari.

6.2.1. Langkah-langkah Percobaan Sequential Search

1. Buatlah Project baru pada dengan nama **TestSearching**
2. Kemudian buat packages baru dengan nama **P7**
3. Buat class **BukuNoAbsen**, kemudian deklarasikan atribut berikut ini:

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

4. Buatlah konstruktor dengan nama **Buku** dengan parameter (**String kodeBuku, String judulBuku, int tahunTerbit, String pengarang, int stock**) kemudian Isi konstruktor tersebut dengan kode berikut!

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

Catatan : Perhatikan konstruktor diatas! Apakah sudah benar? Jika belum, maka perbaiki konstruktor diatas

5. Buatlah method **tampilDataBuku** bertipe void.

```
public void tampilDataBuku(){
    System.out.println("=====");
    System.out.println("Kode buku :"+kodeBuku);
    System.out.println("Judul buku : "+judulBuku);
    System.out.println("Tahun Terbit : "+tahunTerbit);
    System.out.println("Pengarang :"+pengarang);
    System.out.println("Stock : "+stock);
}
```

6. Buat class baru dengan nama **PencarianBukuNoAbsen** seperti di bawah ini!

```
public class PencarianBuku {
    Buku listBk[] = new Buku[5];
    int idx;
}
```

7. Tambahkan method **tambah()** di dalam class tersebut! Method **tambah()** digunakan untuk menambahkan objek dari class Buku ke dalam atribut listBk.

```
void tambah(Buku m) {
    if (idx < listBk.length) {
        listBk[idx] = m;
        idx++;
    } else {
        System.out.println("Data sudah penuh!");
    }
}
```

8. Tambahkan method **tampil()** di dalam class **PencarianBukuNoAbsen** Method **tampil()** digunakan untuk menampilkan semua data buku yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang menggunakan konsep for-each. Syntax tersebut akan memberikan tanda error berupa garis merah, Perbaiki agar tidak ada error!

```
void tampil() {
    for (Buku m : listBk[]) {
        m.tampilDataBuku();
    }
}
```

9. Tambahkan method **FindSeqSearch** bertipe integer dengan parameter **cari** bertipe int. Kemudian Deklarasikan isi method **FindSeqSearch** dengan algoritma pencarian data menggunakan teknik sequential searching.

```

public int FindSeqSearch(int cari) {
    int posisi = 2;
    for (int j = 0; j < listBk.length; j++) {
        if (listBk[j].KodeBuku==cari){
            j = posisi;
            break;
        }
    }
    return posisi;
}

```

10. Buatlah method **Tampilpoisisi** bertipe void dan Deklarasikan isi dari method **Tampilpoisisi**.

```

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

```

11. Buatlah class baru dengan nama **BukuMainNoAbsen** tambahkan method **main**. buatlah sebuah objek PencarianMhs dan buatlah 5 objek buku kemudian tambahkan semua objek buku tersebut dengan memanggil fungsi tambah pada objek PencarianBuku.

```

Scanner s = new Scanner(System.in);
Scanner s1 = new Scanner(System.in);

PencarianBuku data = new PencarianBuku();
int jumBuku = 5;

System.out.println("-----");
System.out.println("Masukkan data Buku secara Urut dari KodeBuku Terkecil : ");
for (int i = 0; i < jumBuku ; i++) {
    System.out.println("-----");
    System.out.print("Kode Buku \t: ");
    int kodeBuku = s.nextInt();
    System.out.print("Judul buku \t : ");
    String judulBuku = s1.nextLine();
    System.out.print("Tahun Terbit \t : ");
    int tahunTerbit= s.nextInt();
    System.out.print("Pengarang \t : ");
    String pengarang= s1.nextLine();
    System.out.print("Stock \t : ");
    int stock= s.nextInt();

    Buku m = new Buku(kodeBuku, judulBuku, tahunTerbit, stock, pengarang);
    data.tambah(m);
}
System.out.println("-----");
System.out.println("Data keseluruhan Mahasiswa : ");

```

Note : perbaiki kode jika terdapat kesalahan.

12. Panggil method `tampil()` untuk melihat semua data yang telah dimasukan.

```
System.out.println("-----");
System.out.println("Data keseluruhan Buku : ");
data.tampil();
```

13. Untuk melakukan pencarian berdasarkan kode buku. Buatlah variable `cari` yang dapat menampung masukan dari keyboard lalu panggil method `FindSeqSearch` dengan isi parameternya adalah variable `cari`. Untuk menampilkan index data yang dicari panggil method `TampilPosisi`

```
System.out.println("_____");
System.out.println("_____");
System.out.println("Pencarian Data : ");
System.out.println("Masukkan Kode Buku yang dicari: ");
System.out.print("Kode Buku : ");
int cari = s.nextInt();
System.out.println("menggunakan sequential Search");
int posisi = data.FindSeqSearch(cari);
data.Tampilposisi(cari, posisi);
```

Verifikasi hasil pencarian :

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
```

```

Data keseluruhan Buku :
=====
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
    
```

Apakah pencarian pada program anda sudah sesuai? Jika belum perbaiki kode sehingga pencarian sesuai.

14. Buatlah method **TampilData** bertipe void pada class **PencarianBukuNoAbsen** dan tambahkan isi dari method **TampilData**.

```

public void TampilData(int x,int pos)
{
    if (pos!= -1) {
        System.out.println("Kode Buku\t : " + x );
        System.out.println("Judul\t : "+listBk[pos].judulBuku);
        System.out.println("Tahun Terbit\t : "+listBk[pos].tahunTerbit);
        System.out.println("Pengarang\t : "+listBk[pos].pengarang);
        System.out.println("Stock\t : "+listBk[pos].stock);
    } else {
        System.out.println("data " + x + "tidak ditemukan");
    }
}
    
```

15. Di dalam method **main()**, Lakukan pemanggilan method **TampilData** dari class **PencarianMhs**.

```

data.TampilData(cari, posisi);
    
```

16. Jalankan dan amati hasilnya.

6.2.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini. Jika hasil belum cocok, perbaiki kode program Anda!

Jika data ditemukan:

```
Pencarian Data :
Masukkan Kode Buku yang dicari:
Kode Buku : 111
menggunakan sequential Search
data : 111 ditemukan pada indeks 0
Kode Buku      : 111
Judul          : Algoritma
Tahun Terbit   : 2019
Pengarang      : Wahyuni
Stock          : 5
```

Jika data tidak ditemukan :

```
Pencarian Data :
Masukkan Kode Buku yang dicari:
Kode Buku : 124
menggunakan sequential Search
data 124 tidak ditemukan
data 124tidak ditemukan
Mamluatuls-MacBook-Air:Praktikum mamluatulhaniah$
```

Note : hasil kode yang ditampilkan diatas adalah hasil pencarian dengan menampilkan index dan data yang dicari

6.2.3. Pertanyaan

1. Jelaskan fungsi **break** yang ada pada method **FindSeqSearch**!

Fungsi break digunakan untuk menghentikan perulangan jika datanya ditemukan.

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? Bisa. Karena perulangan tidak memperhatikan besar kode bukunya.

```
Buku bk1 = new Buku(kodeBuku:105, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Rizky", st...10);
Buku bk2 = new Buku(kodeBuku:102, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st...10);
Buku bk3 = new Buku(kodeBuku:209, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st...10);

pencarian.tambah(bk1);
pencarian.tambah(bk2);
pencarian.tambah(bk3);
```



```
Masukkan kode buku yang dicari:
102
data : 102 ditemukan pada indeks ke-1
Kode Buku: 102
Judul Buku: Pemrograman Python
Tahun Terbit: 2019
Pengarang: Rizky
Stock: 10
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7>
```

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 :

```
Buku dataBuku = data.FindBuku(cari);
dataBuku.tampilDataBuku();
```

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

    return null;
}
```

```
Buku bk1 = new Buku(kodeBuku:105, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Rizky", st...10);
Buku bk2 = new Buku(kodeBuku:102, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st...10);
Buku bk3 = new Buku(kodeBuku:209, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st...10);

pencarian.tambah(bk1);
pencarian.tambah(bk2);
pencarian.tambah(bk3);

// System.out.println("Masukkan kode buku yang dicari: ");
// int cari = sc.nextInt();
// int pos = pencarian.findSeqSearch(cari);
// System.out.println(pos);
// pencarian.tampilPosisi(cari, pos);
// pencarian.tampilData(cari, pos);

Buku dataBuku = pencarian.findBuku(cari:102);
dataBuku.tampilDataBuku();
```

```
an' : \Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7>
Kode Buku: 102
Judul Buku: Pemrograman Python
Tahun Terbit: 2019
Pengarang: Rizky
Stock: 10
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7> █
```

6.3. Searching / Pencarian Menggunakan Binary Search

6.3.1. Langkah-langkah Percobaan Binary Search

1. Pada percobaan 6.2.1 (sequential search) tambahkan method **FindBinarySearch** bertipe integer pada class **PencarianBukuNoAbsen**. Kemudian Deklarasikan isi method **FindBinarySearch** dengan algoritma pencarian data menggunakan teknik binary searching.

```

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;
}

```

2. Panggil method **FindBinarySearch** di kelas **BukuMainNoAbsen**. Kemudian panggil method **tampilposisi** dan **tampilData**

```

System.out.println("=====");
System.out.println("menggunakan binary Search");
posisi = data.FindBinarySearch(cari, 0, jumBuku - 1);
data.Tampilposisi(cari, posisi);
data.TampilData(cari, posisi);

```

3. Jalankan dan amati hasilnya.

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

```
// ...
Buku bk2 = new Buku(kodeBuku:102, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st_10);
Buku bk1 = new Buku(kodeBuku:105, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Rizky", st_10);
Buku bk3 = new Buku(kodeBuku:209, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st_10);

pencarian.tambah(bk1);
pencarian.tambah(bk2);
pencarian.tambah(bk3);
```

```
// Binary Search
int posisi = pencarian.findBinarySearch(cari:209, left:0, 3 - 1);
pencarian.tampilData(x:102, posisi);
```

```
an'
Kode Buku: 209
Judul Buku: Pemrograman Python
Tahun Terbit: 2019
Pengarang: Rizky
Stock: 10
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7> █
```

6.3.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini. Jika hasil belum cocok, **perbaiki kode program Anda!**

hasil kode yang ditampilkan berikut ini adalah hasil pencarian dengan menampilkan index dan data yang dicari. Hasil running penambahan data dan menampilkan data dapat anda lakukan seperti pada percobaan 6.2.1 langkah Nomor 13.

```
=====
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
MamluatulS-MacBook-Air-Praktikum_mamluatulhaniah$
```

6.3.3. Pertanyaan

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

```
mid = (left + right) / 2;
```

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

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

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

```
Buku bk1 = new Buku(kodeBuku:105, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Rizky", st_10);
Buku bk2 = new Buku(kodeBuku:102, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st_10);
Buku bk3 = new Buku(kodeBuku:209, judulBuku:"Pemrograman Python", tahunTerbit:2019, pengarang:"Rizky", st_10);

pencarian.tambah(bk2);
pencarian.tambah(bk3);
pencarian.tambah(bk1);

// System.out.println("Masukkan kode buku yang dicari: ");
```

```
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktikum\16cd9d5e29e44f\redhat.java\jdt_ws\jobsheet
data 105 tidak ditemukan
```

3. 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 seach agar hasilnya sesuai!

Tidak.

```
// // }
pencarian.tambah(new Buku(kodeBuku:20215, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st...10));
pencarian.tambah(new Buku(kodeBuku:20214, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st...10));
pencarian.tambah(new Buku(kodeBuku:20212, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st...10));
pencarian.tambah(new Buku(kodeBuku:20211, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st...10));
pencarian.tambah(new Buku(kodeBuku:20210, judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st...10));
// System.out.println("Masukkan kode buku yang dicari: ");
```

```
// Binary Search
int posisi = pencarian.findBinarySearch(cari:20210, left:0, pencarian.listBk.length - 1);
pencarian.tampilData(x:20210, posisi);
```

```
data 20210 tidak ditemukan
PS C:\Users\Acer\Tugas Kuliah\Semester
```

6.4. Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

1. Buatlah Package baru didalam package P5 dengan nama **MergeSortTest**
2. Tambahkan class **MergeSortingNoAbsen** pada package tersebut
3. Pada class **MergeSortingNoAbsen** buatlah method **mergeSort** yang menerima parameter data array yang akan diurutkan

```
public void mergeSort(int[] data) {
```

4. Buatlah method **merge** untuk melakukan proses penggabungan data dari bagian kiri dan kanan.



```
public void merge(int data[], int left, int middle, int right) {
```

- 5 Implementasikan proses **merge** sebagai berikut.



```
int[] temp = new int[data.length];
for (int i = left; i <= right; i++) {
    temp[i] = data[i];
}
int a = left;
int b = middle + 1;
int c = left;

while (a <= middle && b <= right) {
    if (temp[a] <= temp[b]) {
        data[c] = temp[a];
        a++;
    } else {
        data[c] = temp[b];
        b++;
    }
    c++;
}
int s = middle - a;
for (int i = 0; i <= s; i++) {
    data[c + i] = temp[a + i];
}
```

- 6 Buatlah method **sort**



```
public void sort(int data[], int left, int right) {
```

- 7 Implementasikan kode berikut pada method **sort**



```
if (left < right) {
    int middle = (left + right) / 2;
    sort(data, left, middle);
    sort(data, middle + 1, right);
    merge(data, left, middle, right);
}
```

- 8 Pada method **mergeSort**, panggil method **sort** dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir.
- 9 Tambahkan method **printArray**

```

public void printArray(int arr[]) {
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
    System.out.println();
}

```

- 10 Sebagai langkah terakhir, deklarasikan data yang akan diurutkan kemudian panggil proses sorting pada class **MergeSortMainNoAbsen**. Tambahkan fungsi main pada kelas tersebut, kemudian tuliskan kode berikut didalam fungsi main.

```

int data[] = {10,40,30,50,70,20,100,90};
System.out.println("sorting dengan merge sort");
MergeSort mSort= new MergeSort();
System.out.println("data awal");
mSort.printArray(data);
mSort.mergeSort(data);
System.out.println("setelah diurutkan");
mSort.printArray(data);

```

6.4.2. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```

sorting dengan merge sort
data awal
10 40 30 50 70 20 100 90
setelah diurutkan
10 20 30 40 50 70 90 100

```

```

PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7> & 'C:\
\Code\User\workspaceStorage\c202c9eaa23e016e0816cd9d5e29e44f\redhat.java\jdt_w
Data sebelum diurutkan
10 40 30 40 70 20 100 90
Data setelah diurutkan
10 20 30 40 40 70 90 100
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7>

```

6.5. Latihan Praktikum

Tim Ajar Algoritma dan Struktur Data 2023-2024

Jurusan Teknologi Informasi-Politeknik Negeri Malang

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.

```
public int findSeqSearch(String cari) {
    int posisi = -1;

    for (int i = 0; i < listBk.length; i++) {
        if (listBk[i].kodeBuku.equals(cari)) {
            posisi = i;
            break;
        }
    }

    return posisi;
}

public int findBinarySearch(String cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (left + right) / 2;
        // Membandingkan String dengan metode compareTo()
        if (listBk[mid].kodeBuku.compareTo(cari) == 0) {
            return mid;
        } else if (listBk[mid].kodeBuku.compareTo(cari) > 0) {
            return findBinarySearch(cari, left, mid - 1);
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
}
```

Run | Debug

```
public static void main(String[] args) {
```

```
    Pencarian pencarian = new Pencarian();
```

```
    pencarian.tambah(new Buku(kodeBuku:"20210", judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st_10));
```

```
    pencarian.tambah(new Buku(kodeBuku:"20212", judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st_10));
```

```
    pencarian.tambah(new Buku(kodeBuku:"20215", judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st_10));
```

```
    pencarian.tambah(new Buku(kodeBuku:"20214", judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st_10));
```

```
    pencarian.tambah(new Buku(kodeBuku:"20211", judulBuku:"Pemrograman Java", tahunTerbit:2019, pengarang:"Andi", st_10));
```

```
    System.out.println(x:"==== Sequential Search =====");
```

```
    int pos = pencarian.findSeqSearch(cari:"20215");
```

```
    pencarian.tampilPosisi(x:"20215", pos);
```

```
    pencarian.tampilData(x:"20215", pos);
```

```
    System.out.println(x:"==== Binary Search =====");
```

```
    pos = pencarian.findBinarySearch(cari:"20215", left:0, pencarian.listBk.length - 1);
```

```
    pencarian.tampilPosisi(x:"20215", pos);
```

```
    pencarian.tampilData(x:"20215", pos);
```

```
}
```

```
Tahun Terbit: 2019
```

```
& 'C:\Program Files\Zulu\zu1
```

```
16cd9d5e29e44f\redhat.java\jdt_ws\jobsheet 7_7803f0a9\bin' 'mergeShort.Pencarian' S C:\Users\Acer\Tug
```

```
==== Sequential Search =====
```

```
data : 20215 ditemukan pada indeks ke-2
```

```
Kode Buku: 20215
```

```
Judul Buku: Pemrograman Java
```

```
Tahun Terbit: 2019
```

```
Pengarang: Andi
```

```
Stock: 10
```

```
==== Binary Search =====
```

```
data : 20215 ditemukan pada indeks ke-2
```

```
Kode Buku: 20215
```

```
Judul Buku: Pemrograman Java
```

```
Tahun Terbit: 2019
```

```
Pengarang: Andi
```

```
Stock: 10
```

```
PS C:\Users\Acer\Tugah Kuliah\Semester 2\Praktek Algoritma\jobsheet 7> █
```

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
- 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!

```

1  package mergeShort;
2
3  import java.util.Scanner;
4
5  public class Pencarian {
6      Buku listBk[] = new Buku[5];
7      int idx;
8
9      void tambah(Buku bk) {
10         if (idx < listBk.length) {
11             listBk[idx] = bk;
12             idx++;
13         } else {
14             System.out.println("Data sudah penuh!");
15         }
16     }
17
18     public int findSeqSearch(String cari) {
19         int posisi = -1;
20
21         for (int i = 0; i < listBk.length; i++) {
22             if (listBk[i].kodeBuku.equals(cari)) {
23                 posisi = i;
24                 break;
25             }
26         }
27
28         return posisi;
29     }
30
31     public int findBinarySearch(String cari, int left, int right) {
32         int mid;
33         if (right >= left) {
34             mid = (left + right) / 2;
35             // Membandingkan String dengan metode compareTo()
36             if (listBk[mid].kodeBuku.compareTo(cari) == 0) {
37                 return mid;
38             } else if (listBk[mid].kodeBuku.compareTo(cari) > 0) {
39                 return findBinarySearch(cari, left, mid - 1);
40             } else {
41                 return findBinarySearch(cari, mid + 1, right);
42             }
43         }
44         return -1;
45     }
46
47     public int findSeqSearchByJudul(String cari) {
48         int foundIndex = -1;
49         boolean multipleFound = false;
50
51         for (int i = 0; i < listBk.length; i++) {
52             if (listBk[i].judulBuku.equals(cari)) {
53                 if (foundIndex == -1) {
54                     foundIndex = i;
55                 } else {
56                     multipleFound = true;
57                     break;
58                 }
59             }
60         }
61
62         if (multipleFound) {
63             System.out.println("Buku " + cari + " ada lebih dari satu di index berikut:");
64             for (int i = 0; i < listBk.length; i++) {
65                 if (listBk[i].judulBuku.equals(cari)) {
66                     System.out.print(i + " ");
67                 }
68             }
69             System.out.println();
70         } else if (foundIndex == -1) {
71             System.out.println("Buku " + cari + " tidak ditemukan");
72         } else {
73             System.out.println("Buku " + cari + " ditemukan di index " + foundIndex);
74         }
75
76         return foundIndex;
77     }
78 }
79

```

```

79
80 public int findBinarySearchByJudul(String cari, int left, int right) {
81     int foundIndex = -1;
82
83     while (left <= right) {
84         int mid = (left + right) / 2;
85
86         if (listBk[mid].judulBuku.compareTo(cari) == 0) {
87             foundIndex = mid;
88             int temp = mid - 1;
89
90             while (temp >= left && listBk[temp].judulBuku.equals(cari)) {
91                 foundIndex = temp;
92                 temp--;
93             }
94
95             break;
96         } else if (listBk[mid].judulBuku.compareTo(cari) > 0) {
97             right = mid - 1;
98         } else {
99             left = mid + 1;
100         }
101     }
102
103     if (foundIndex == -1) {
104         System.out.println("Buku " + cari + " tidak ditemukan");
105     } else if (listBk[foundIndex].judulBuku.equals(cari)) {
106         System.out.println("Buku " + cari + " ditemukan di index " + foundIndex);
107     } else {
108         System.out.println("Judul yang cocok tidak ditemukan (mungkin ada duplikat dengan judul berbeda)");
109     }
110
111     return foundIndex;
112 }
113
114
115
116
117 public void tampilPosisi(String x, int pos) {
118     if (pos != -1) {
119         System.out.println("data : " + x + " ditemukan pada indeks ke-" + pos);
120     } else {
121         System.out.println("data " + x + " tidak ditemukan");
122     }
123 }
124
125 public void tampilData(String x, int pos) {
126     if (pos != -1) {
127         listBk[pos].tampilDataBuku();
128     } else {
129         System.out.println("data " + x + " tidak ditemukan");
130     }
131 }
132
133
134 public void mergeSort(Buku[] data) {
135     sort(data, 0, data.length - 1);
136 }
137
138 public void sort(Buku[] data, int left, int right) {
139     if (left < right) {
140         int middle = (left + right) / 2;
141         sort(data, left, middle);
142         sort(data, middle + 1, right);
143         merge(data, left, middle, right);
144     }
145 }
146
147 public void merge(Buku[] data, int left, int middle, int right) {
148     // Create temporary array to hold sorted elements
149     Buku[] temp = new Buku[data.length];
150
151     // Copy data to temporary array
152     for (int i = left; i <= right; i++) {
153         temp[i] = data[i];
154     }
155
156     int a = left;
157     int b = middle + 1;
158     int c = left;
159
160     while (a <= middle && b <= right) {
161         // Compare elements based on kodeBuku using compareTo()
162         if (data[a].judulBuku.compareTo(data[b].judulBuku) <= 0) {
163             data[c] = temp[a];
164             a++;
165         } else {
166             data[c] = temp[b];
167             b++;
168         }
169         c++;
170     }
171
172     // Copy remaining elements from temporary array
173     int s = middle - a;
174     for (int i = 0; i <= s; i++) {
175         data[c + i] = temp[a + i];
176     }
177 }

```

```

177     }
178
179     public static void main(String[] args) {
180
181         Pencarian pencarian = new Pencarian();
182
183         pencarian.tambah(new Buku("20210", "Habis Gelap Terbitlah Terang", 2019, "Andi", 10));
184         pencarian.tambah(new Buku("20212", "Dilan", 2019, "Andi", 10));
185         pencarian.tambah(new Buku("20215", "Pemrograman Java", 2019, "Andi", 10));
186         pencarian.tambah(new Buku("20214", "Pemrograman Javascript", 2019, "Andi", 10));
187         pencarian.tambah(new Buku("20211", "Pemrograman Java", 2019, "Andi", 10));
188
189         pencarian.mergeSort(pencarian.listBk);
190
191         // tampilkan data setelah sort
192         for (int i = 0; i < pencarian.listBk.length; i++) {
193             System.out.println("Judul: " + pencarian.listBk[i].judulBuku);
194         }
195
196         // search By judul buku
197         System.out.println("==== Sequential Search =====");
198         int pos = pencarian.findSeqSearchByJudul("Pemrograman Java");
199         pencarian.tampilPosisi("Pemrograman Java", pos);
200         pencarian.tampilData("Pemrograman Java", pos);
201
202         System.out.println("==== Binary Search =====");
203         pos = pencarian.findBinarySearchByJudul("Pemrograman Java", 0, pencarian.listBk.length - 1);
204         pencarian.tampilPosisi("Pemrograman Java", pos);
205         pencarian.tampilData("Pemrograman Java", pos);
206
207
208         // // search By kode buku
209         // System.out.println("==== Sequential Search =====");
210         // int pos = pencarian.findSeqSearch("20215");
211         // pencarian.tampilPosisi("20215", pos);
212         // pencarian.tampilData("20215", pos);
213
214         // System.out.println("==== Binary Search =====");
215         // pos = pencarian.findBinarySearch("20215", 0, pencarian.listBk.length - 1);
216         // pencarian.tampilPosisi("20215", pos);
217         // pencarian.tampilData("20215", pos);
218
219     }
220 }
221

```

```

aceStorage\c202c9eaa23e016e0816cd9d5e29e44f\redhat.java\jdt_ws\jobsheet 7_780
Judul: Dilan
Judul: Habis Gelap Terbitlah Terang
Judul: Pemrograman Java
Judul: Pemrograman Java
Judul: Pemrograman Javascript
===== Sequential Search =====
Buku Pemrograman Java ada lebih dari satu di index berikut:
data : Pemrograman Java ditemukan pada indeks ke-2
Kode Buku: 20215
Judul Buku: Pemrograman Java
Tahun Terbit: 2019
Pengarang: Andi
Stock: 10
===== Binary Search =====
Buku Pemrograman Java ditemukan di index 2
data : Pemrograman Java ditemukan pada indeks ke-2
Kode Buku: 20215
Judul Buku: Pemrograman Java
Tahun Terbit: 2019
Pengarang: Andi
Stock: 10
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 7>

```