**Nama: Rizqi Bagus Andrean**

**Kelas: TI-1D  
Absen: 25**

JOBSHEET VI SEARCHING

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.

# Searching / Pencarian Menggunakan Agoritma 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 berfunsi 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 |

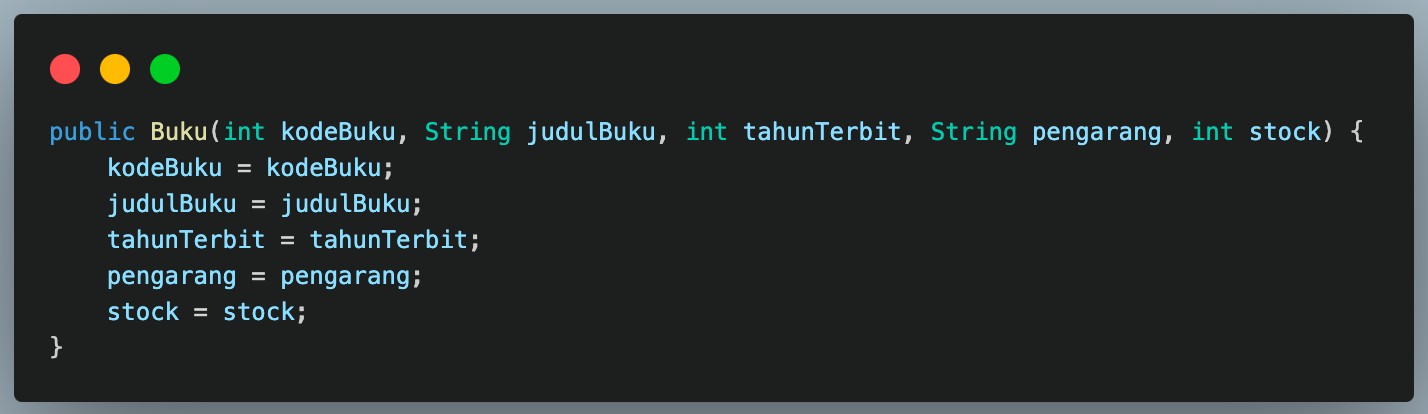
Selanjutnya class diagram PencarianBuku merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array Buku, misalkan untuk menambahkan objek Buku, menampilkan semua data Buku, untuk melakukan pencarian berdasarkan kode Buku menggunakan algoritma Sequential Search, menampilkan posisi dari data yang dicari, serta menampilkan data Buku yang dicari.

# Langkah-langkah Percobaan Sequential Search

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



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



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

1. Buatlah method **tampilDataBuku** bertipe void.
2. Buat class baru dengan nama **PencarianBukuNoAbsen** seperti di bawah ini!
3. Tambahkan method **tambah()** di dalam class tersebut! Method **tambah()** digunakan untuk menambahkan objek dari class Buku ke dalam atribut listBk.



1. 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!



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



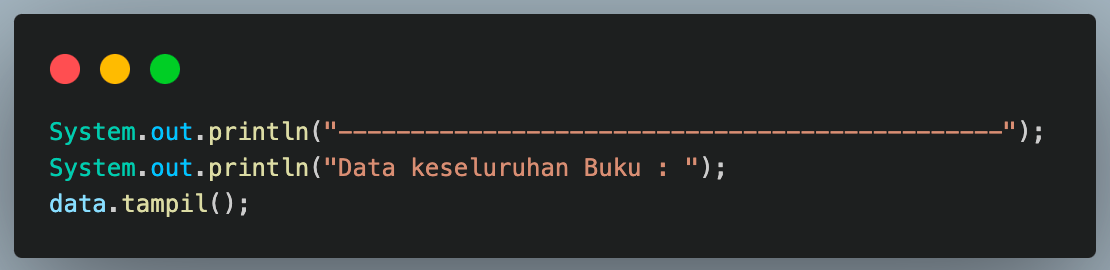
1. Buatlah method **Tampilpoisisi** bertipe void dan Deklarasikan isi dari method

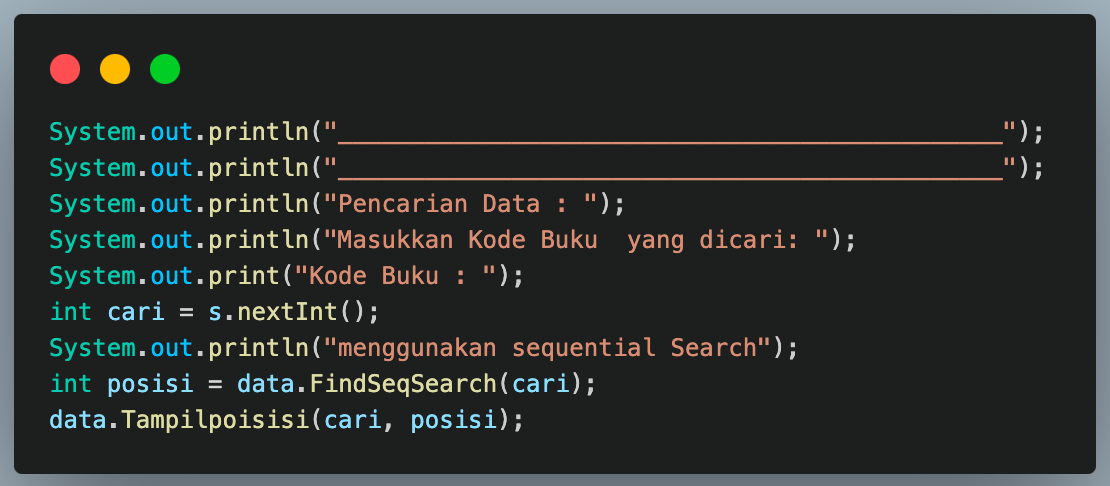
**Tampilpoisisi.**

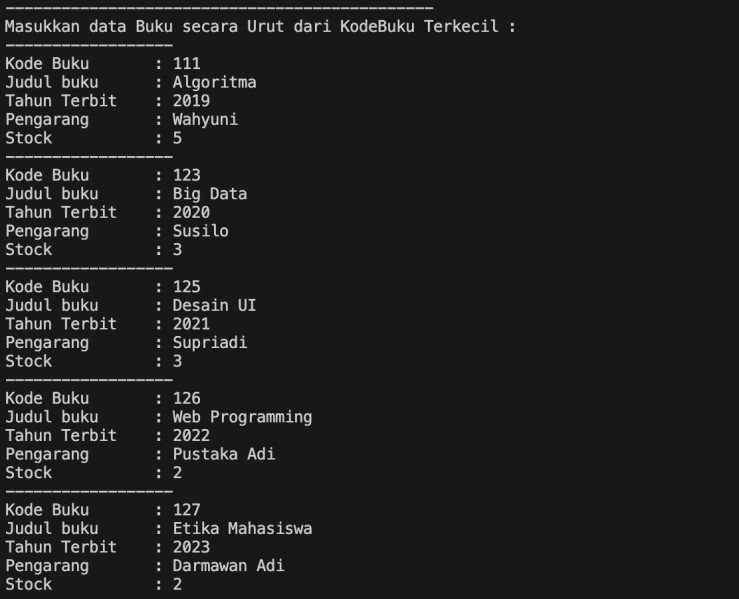
1. Buatlah class baru dengan nama **BukuMainNoAbsen** tambahkan method **main**. buatlah sebuah objek PencarianMhs dan buatlah 5 objek buku kemudian tambahkan semua objek buku tersebut dengan memanggil fungsi tambah pada objek PencarianBuku.

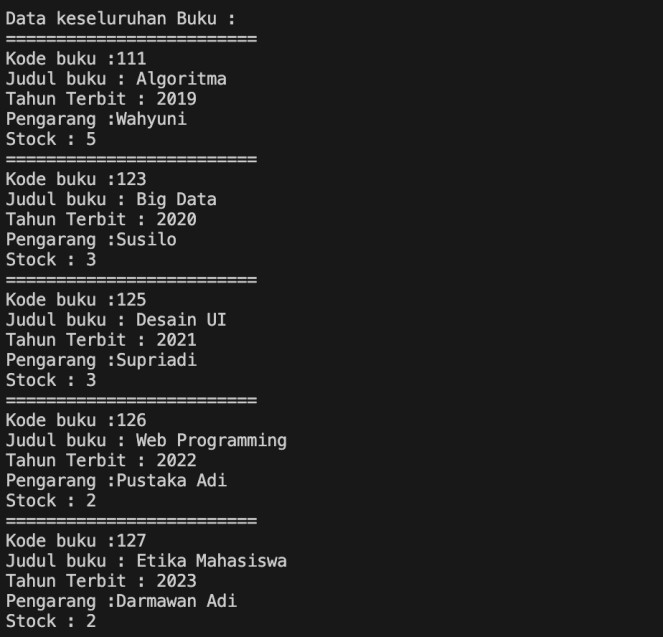


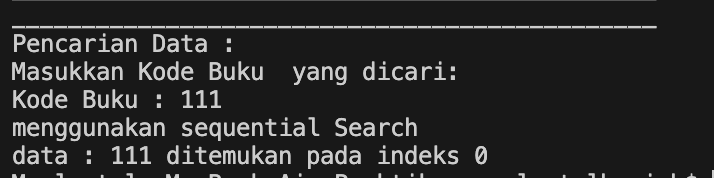
Note : perbaiki kode jika terdapat kesalahan.

1. Panggil method **tampil()** untuk melihat semua data yang telah dimasukan.
2. 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



Verifikasi hasil pencarian :





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

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



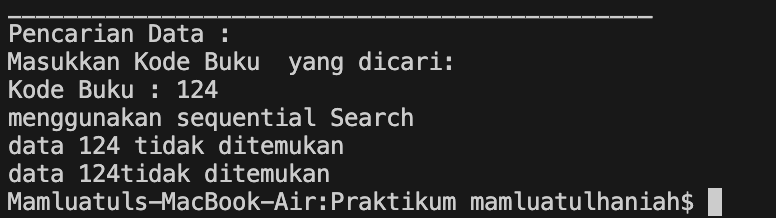
1. Di dalam method **main(),** Lakukan pemanggilan method **TampilData** dari class

**PencarianMhs.**

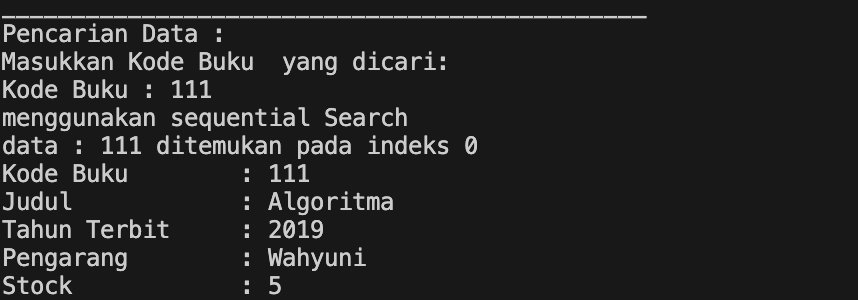
1. Jalankan dan amati hasilnya.

# Verifikasi Hasil Percobaan

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



Jika data ditemukan:



Jika data tidak ditemukan :

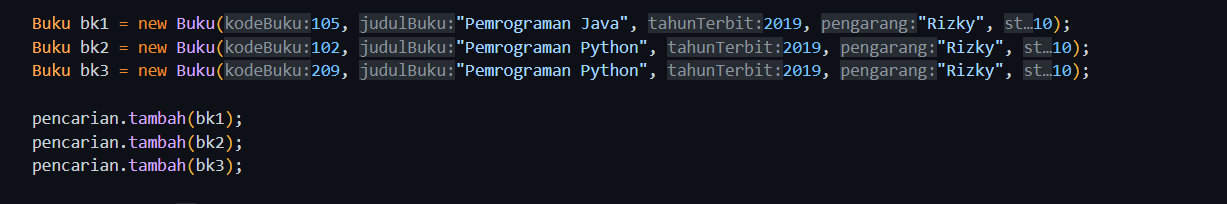
Note : hasil kode yang ditampilkan diatas adalah hasil pencarian dengan menampilkan index dan data yang dicari

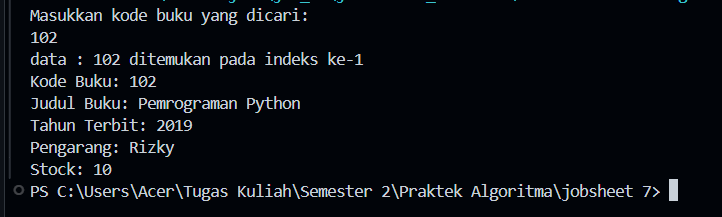
# Pertanyaan

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

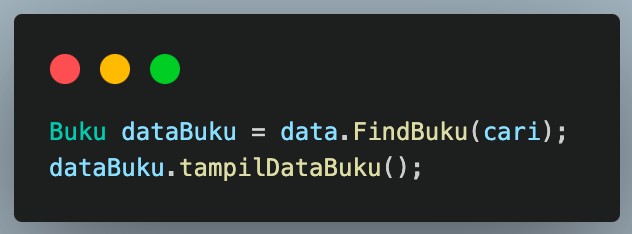
**Fungsi break digunakan untuk menghentikan perulangan jika datanya ditemukan.**

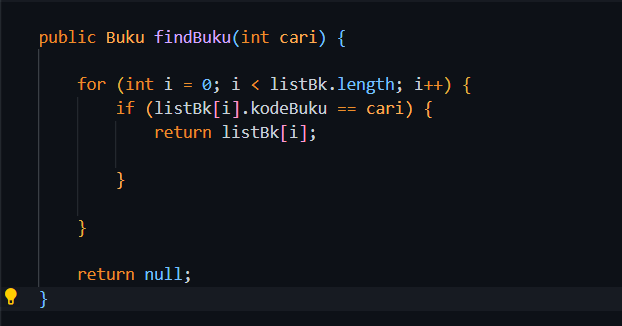
1. Jika Data Kode Buku yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Tunjukkan hasil screenshoot untuk bukti dengan kode Buku yang acak. Jelaskan Mengapa hal tersebut bisa terjadi? Bisa. Karena perulangan tidak memperhatikan besar kode bukunya.

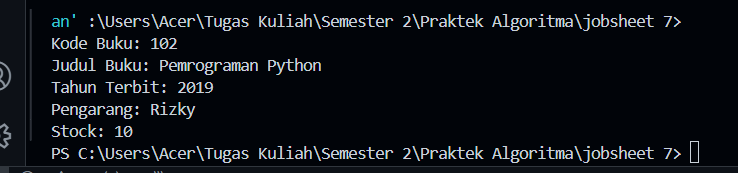




1. Buat method baru dengan nama **FindBuku** menggunakan konsep sequential search dengan tipe method dari **FindBuku** adalah **BukuNoAbsen**. Sehingga Anda bisa memanggil method tersebut pada class **BukuMain** seperti gambar berikut :



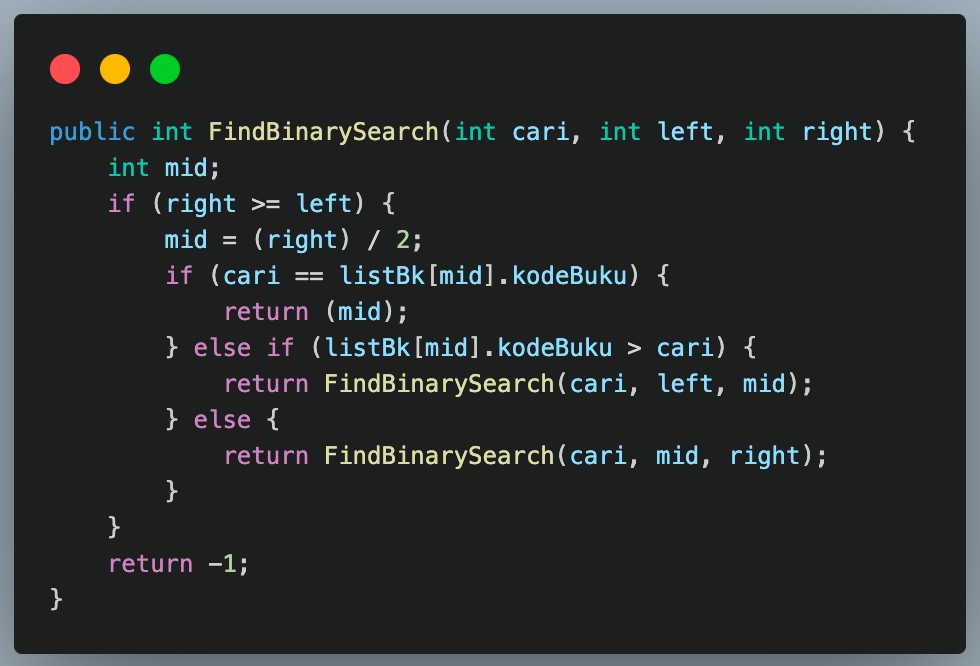




# Searching / Pencarian Menggunakan Binary Search

* + 1. **Langkah-langkah Percobaan Binary Search**

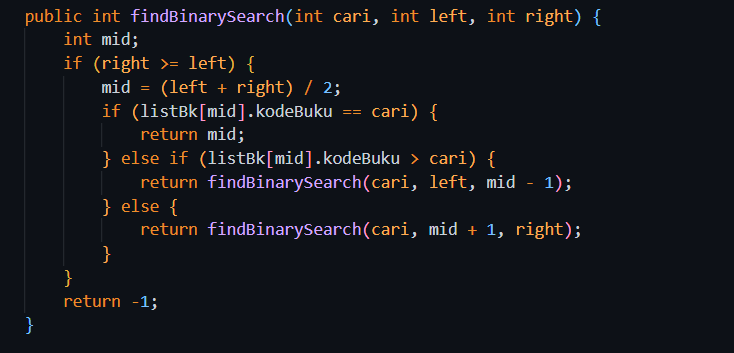
1. Pada percobaan 6.2.1 (sequential search) tambahkan method **FindBinarySearch** bertipe integer pada class **PencarianBukuNoAbsen**. Kemudian Deklarasikan isi method **FindBinarySearch** dengan algoritma pencarian data menggunakan teknik binary searching.

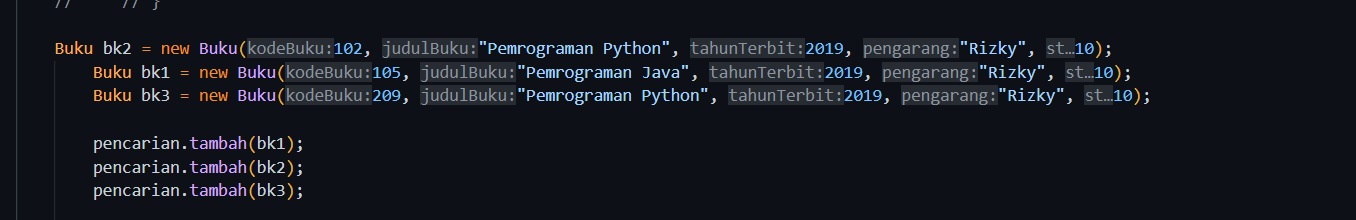


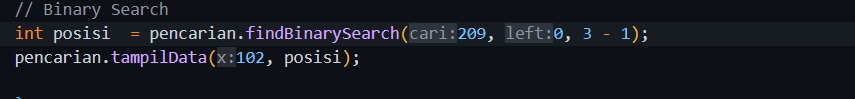
1. Panggil method **FindBinarySearch** di kelas **BukuMainNoAbsen.** Kemudia panggil

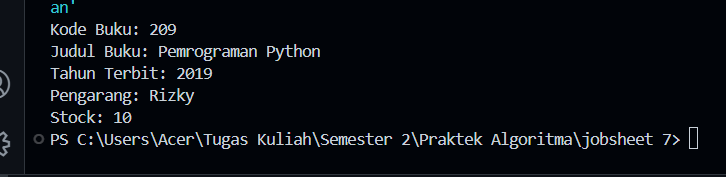
## A screen shot of a computer code Description automatically generatedmethod tampilposisi dan tampilData

1. Jalankan dan amati hasilnya.





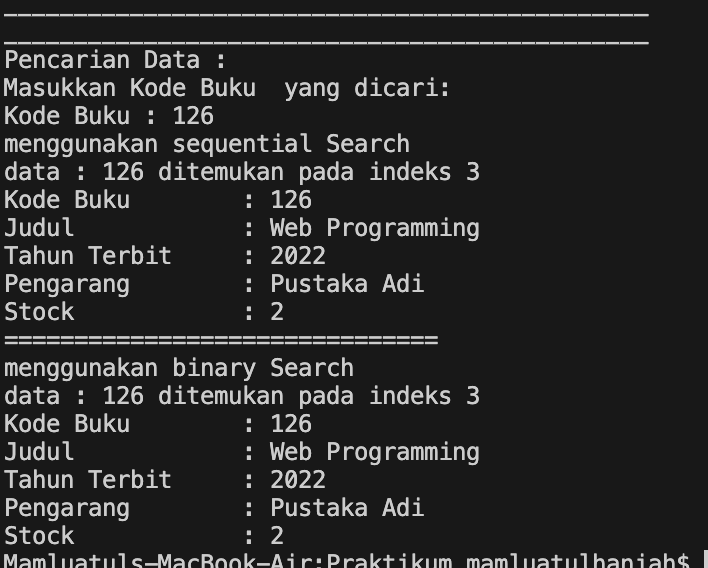




# Verifikasi Hasil Percobaan

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.

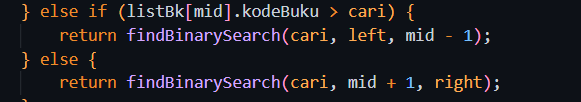


# Pertanyaan

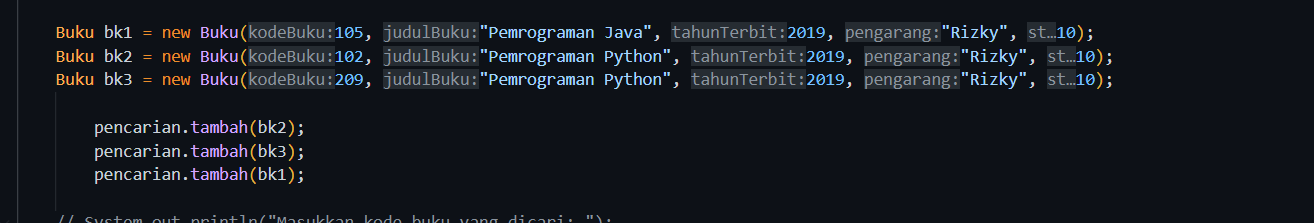
* + - 1. Tunjukkan pada kode program yang mana proses divide dijalankan!

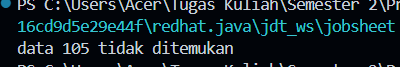


* + - 1. Tunjukkan pada kode program yang mana proses conquer dijalankan!



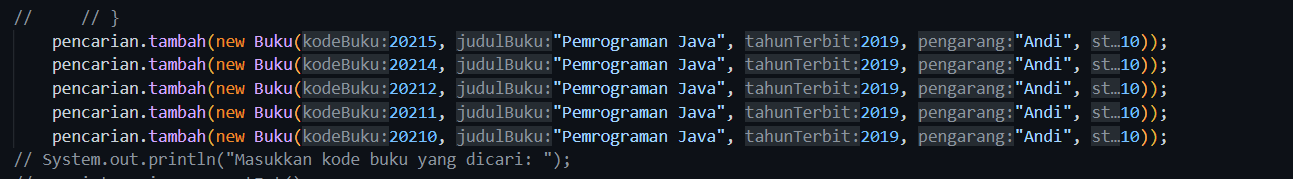
4. Jika data Kode Buku yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian! Tunjukkan hasil screenshoot untuk bukti dengan kode Buku yang acak. Jelaskan Mengapa hal tersebut bisa terjadi? Tidak

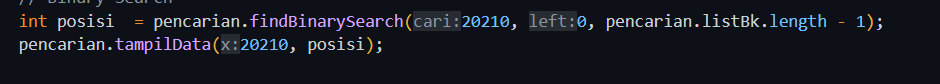




* + - 1. Jika Kode Buku yang dimasukkan dari Kode Buku terbesar ke terkecil (missal : 20215, 20214, 20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai!

Tidak.







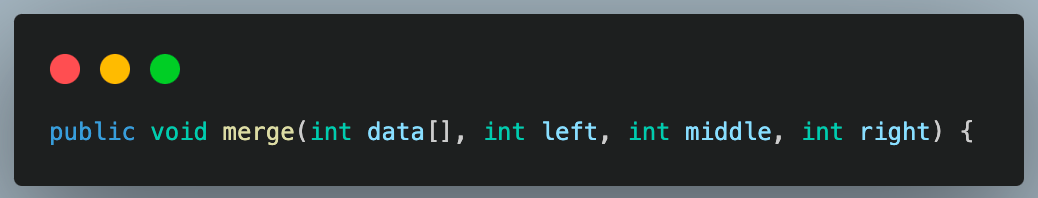
# Percobaan Pengayaan Divide and Conquer

* + 1. **Langkah-langkah Percobaan Merge Sort**

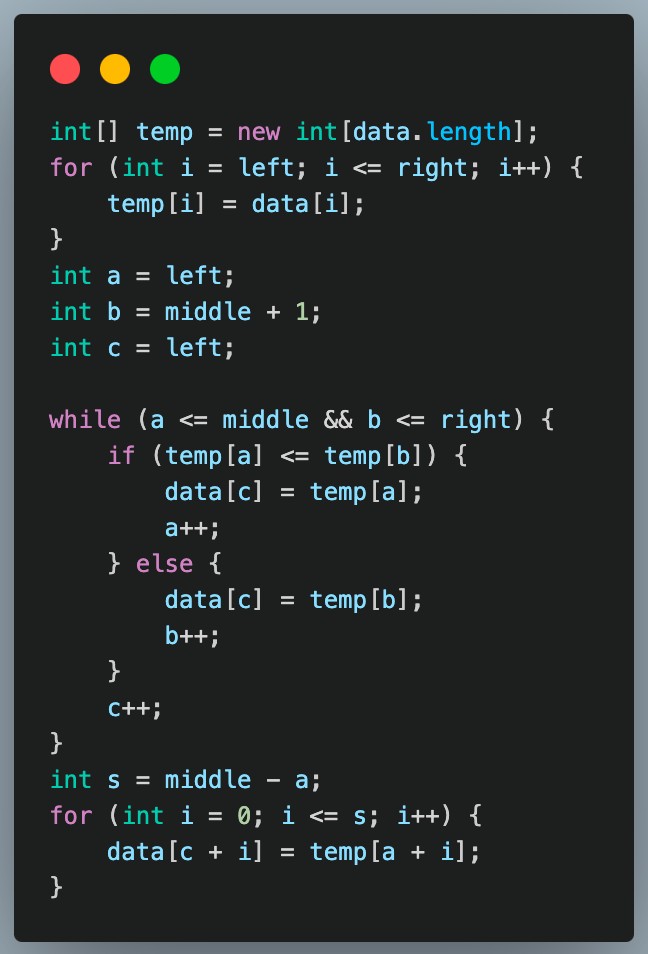
1. Buatlah Package baru didalam package P5 dengan nama **MergeSortTest**
2. Tambahkan class **MergeSortingNoAbsen** pada package tersebut
3. Pada class **MergeSortingNoAbsen** buatlah method **mergeSort** yang menerima parameter data array yang akan diurutkan



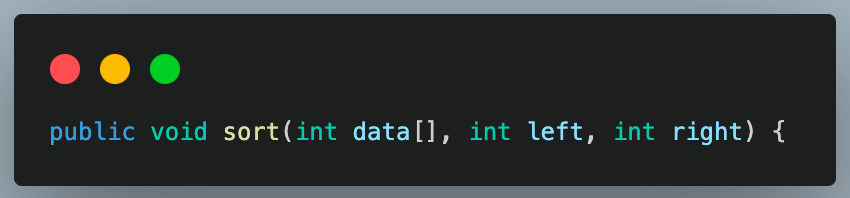
1. Buatlah method **merge** untuk melakukan proses penggabungan data dari bagian kiri dan kanan.



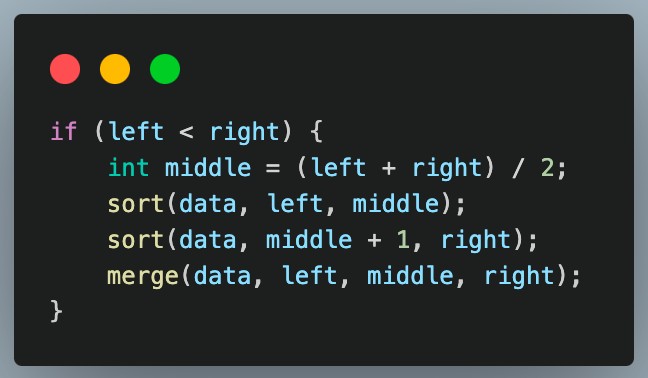
1. Implementasikan proses **merge** sebagai berikut.



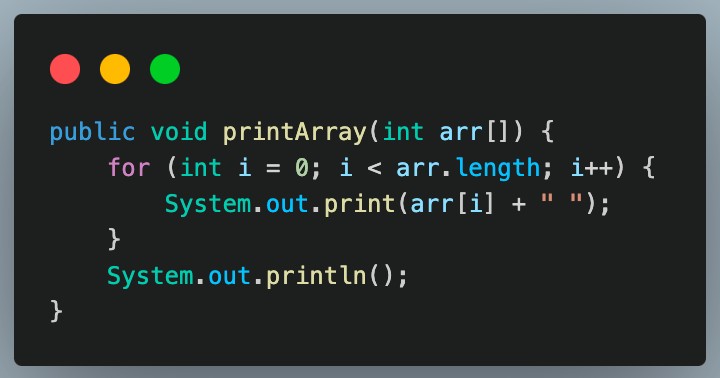
1. Buatlah method **sort**



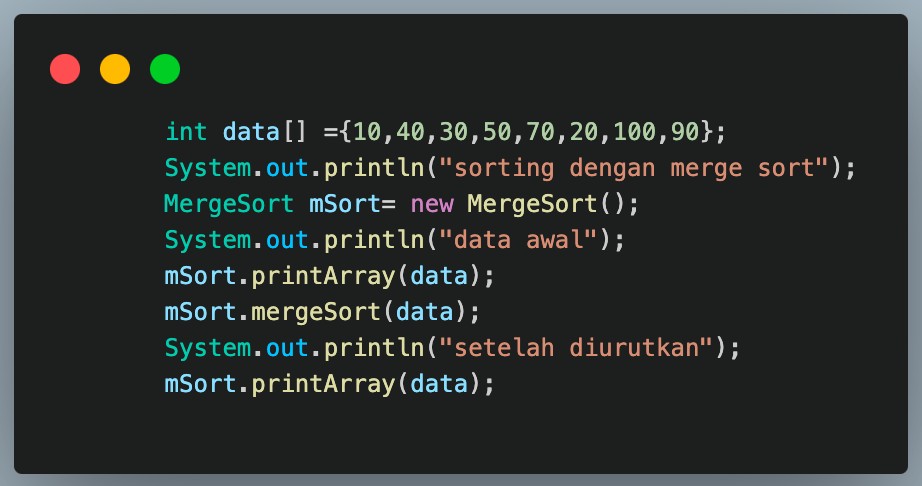
1. Implementasikan kode berikut pada method **sort**



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

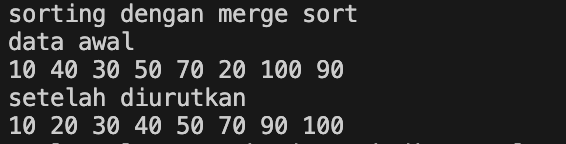


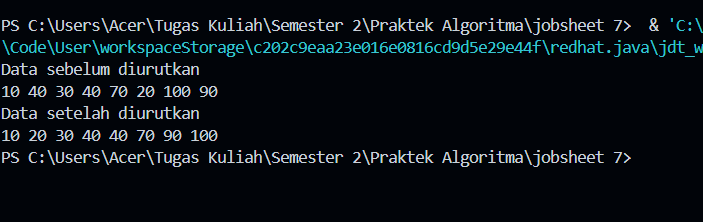
1. Sebagai langkah terakhir, deklarasikan data yang akan diurutkan kemudian panggil proses sorting pada class **MergeSortMainNoAbsen.** Tambahkan fungsi main pada kelas tersebut, kemudian tuliskan kode berikut didalam fungsi main.



## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



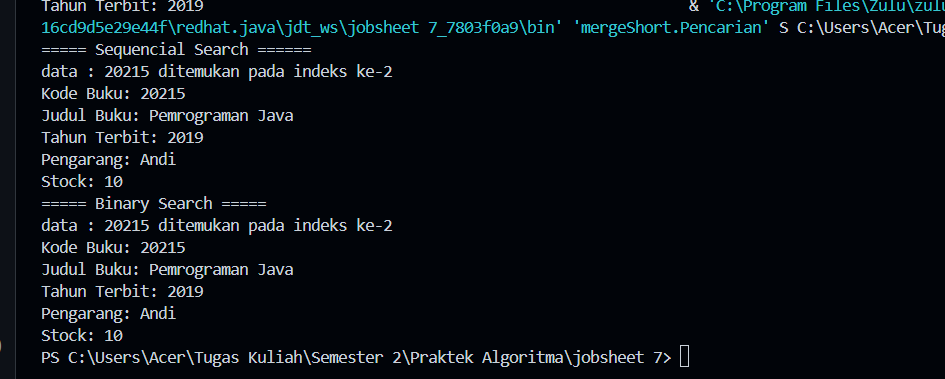


# Latihan Praktikum

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.







1. 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!







