

NAMA : FERNANDO DJAKA S.P
NIM/KELAS : V3922022/TI.D

PENJELASAN KODE PROGRAM :

```
import java.util.ArrayList;  
import java.util.List;
```

Pada baris-baris pertama, mengimpor kelas 'ArrayList' dan 'List' dari paket 'java.util'. Ini diperlukan untuk menggunakan struktur data 'ArrayList' dan 'List' yang akan digunakan dalam implementasi 'BookManager'

```
public class Main {  
    public static void main(String[] args) {  
        // ...  
    }
```

mendefinisikan kelas utama 'Main' yang berfungsi sebagai titik masuk program. Kode program yang berada di dalam metode 'main' akan dieksekusi ketika program dijalankan.

```
public static class BookManager {  
    // ...  
}
```

mendefinisikan kelas 'BookManager', yang bertanggung jawab atas manajemen buku-buku. Kelas ini memiliki atribut 'books' yang bertipe 'List<Book>' untuk menyimpan daftar buku. Kelas ini juga memiliki metode 'addBook(Book book)' untuk menambahkan buku baru ke dalam daftar, metode 'displayBooks()' untuk menampilkan daftar buku, dan metode 'searchBook(String title)' untuk mencari buku berdasarkan judul.

```
Book book1 = new Book("Java Programming", "FERNANDO DJAKA", 2020);  
Book book2 = new Book("Python Basics", "Masih saya sendiri", 2019);
```

membuat dua objek buku menggunakan konstruktor 'Book' dengan detail yang berbeda. Dengan membuat objek buku, kita mengaplikasikan konsep pemrograman berorientasi objek dengan menggunakan kelas 'Book' sebagai cetak biru (blueprint) untuk menciptakan objek konkret.

```
BookManager bookManager = new BookManager();
```

membuat objek 'BookManager' menggunakan konstruktor 'BookManager'. mengaplikasikan konsep pemrograman berorientasi objek dengan menggunakan kelas 'BookManager' sebagai pengelola objek buku.

```
bookManager.addBook(book1);  
bookManager.addBook(book2);
```

menambahkan objek buku 'book1' dan 'book2' ke 'BookManager' dengan menggunakan metode 'addBook(Book book)'. Dengan ini, kita mengaplikasikan konsep enkapsulasi dan pengelolaan objek melalui kelas 'BookManager'.

```
bookManager.displayBooks();
```

Kita menampilkan daftar buku yang ada di 'BookManager' dengan menggunakan metode 'displayBooks()'. Dengan ini, kita mengaplikasikan konsep pemrograman berorientasi objek dengan memanggil metode pada objek 'BookManager' untuk melakukan tugas-tugas yang terkait.

```
String searchTitle = "Java Programming";  
Book foundBook = bookManager.searchBook(searchTitle);  
if (foundBook != null) {  
    // ...  
} else {  
    // ...  
}
```

melakukan pencarian buku berdasarkan judul dengan menggunakan metode 'searchBook(String title)' pada objek 'BookManager'. Hasil pencarian disimpan dalam variabel 'foundBook'. Dalam blok kondisional 'if-else', kita mengecek apakah buku ditemukan atau tidak, dan menampilkan informasi buku jika ditemukan atau pesan "Book not found." jika tidak ditemukan. Dengan ini, kita mengaplikasikan konsep pemrograman berorientasi objek dengan menggunakan metode pada objek 'BookManager' untuk mendapatkan informasi yang dibutuhkan.

Melalui penerapan konsep-konsep pemrograman berorientasi objek seperti enkapsulasi, pewarisan, dan pengelolaan objek melalui kelas, proyek ini memperlihatkan penggunaan metode pemrograman berorientasi objek dalam bahasa pemrograman Java.