**Laporan Proyek**

SISTEM LIBRARY



**1132102 - Object Oriented Programming**

**Disusun Oleh :**

**11322029 Carlos Mapaung**

**11322034 Joi Diego Napitupulu**

**11322039 Kenan Tomfie Bukit**

**11322024 Immanuel Partogi Pardede**

**Diploma III Teknologi Informasi IT Del 2023**

**Sitoluama, Laguboti 2023**

1. **FUNCTIONAL DESCRIPTION**

Pada sub bab ini akan dijelaskan fungsi-fungsi utama pada sistem *libary*, beberapa fungsi dalam sistem *library* yang dapat dikerjakan *User* antara lain:

* + - 1. Fungsi *login*

Fungsi *login* digunakan oleh *user* untuk dapat masuk dan mengakses ke dalam sistem dengan memasukkan *student number* dan password user.

1. Fungsi membaca dan melihat list buku

Fungsi ini dilakukan oleh *user* untuk mengakses dan melihat informasi buku yang sudah ditambahkan ke dalam sistem.

1. Fungsi menambahkan Peminjaman Buku

Fungsi ini dilakukan oleh *user untuk* menambahkan list peminjaman buku. Adapun data yang harus harus diisi pada form adalah *first name, last name, gender* dan *book title* dari *user.*

1. Fungsi menambahkan WistList Buku

Fungsi ini dilakukan oleh *user* untuk menambahkan daftar buku yang ingin dipinjam di kemudian hari.

1. Fungsi pengembalian Buku

Fungsi ini dilakukan oleh *user* untuk dapat memperbarui data pengembalian buku yang telah dipinjam sebelumnya.

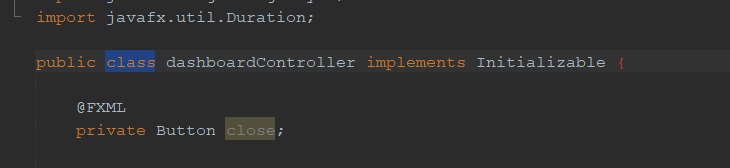
1. Fungsi mengubah dan menambah Foto Profil

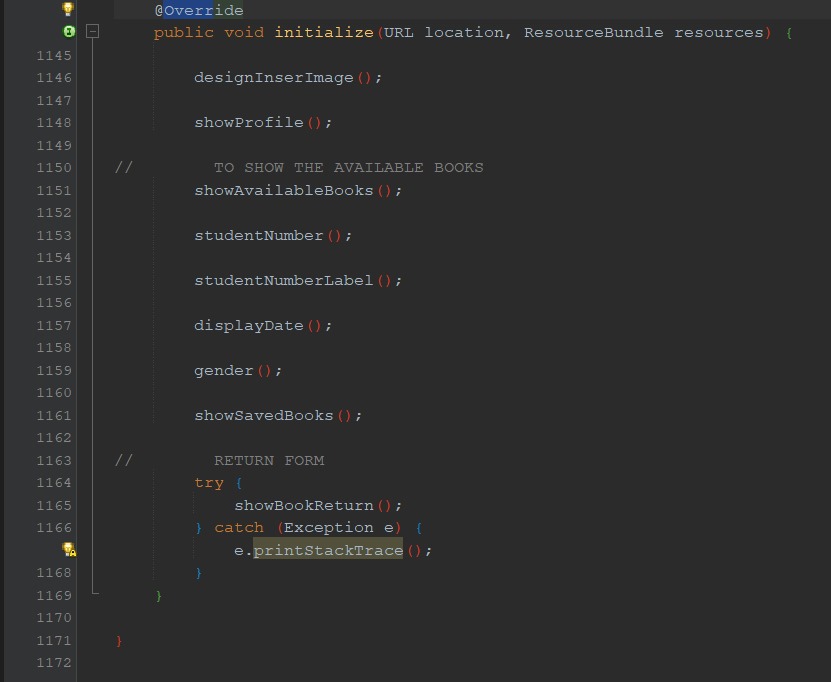
Fungsi ini dilakukan oleh *user* untuk dapat mengubah dan menambahkan foto profil pada halaman *dashboard user.*

1. **OOP IMPLEMENTATION**
   1. **Abstraction**

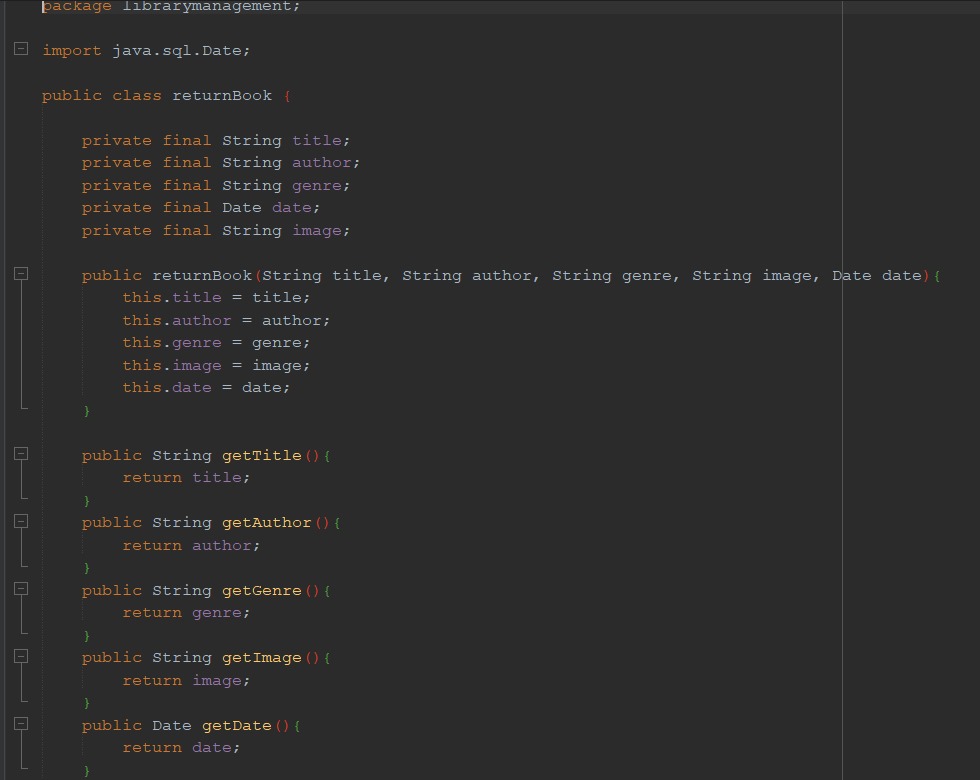
Dalam OOP, abstraksi adalah konsep yang memungkinkan kita untuk menyembunyikan detail implementasi dari suatu objek dan hanya mengekspos fungsionalitas yang relevan atau yang dapat diakses oleh pengguna objek.

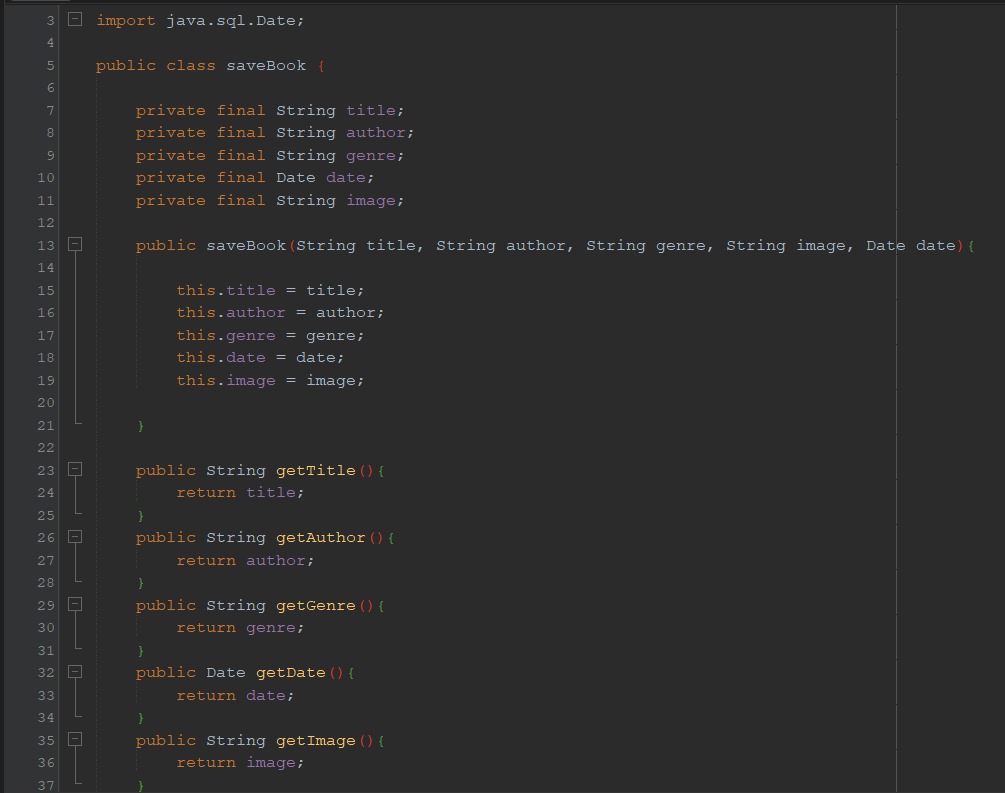
Contoh penggunaan abstraksi dalam OOP dapat dilihat dalam pembuatan kelas dashboardController yang mengimlementasikan class Initializable





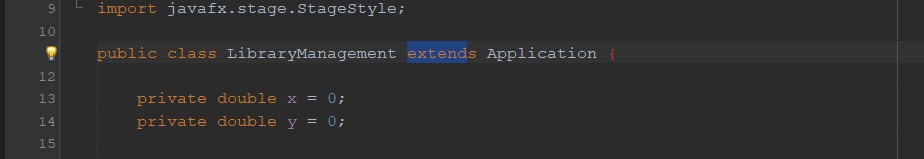
* 1. **Encapsulation**

Encapsulation adalah mekanisme dalam pemrograman berorientasi objek (OOP) yang menggabungkan data dan perilaku dalam satu unit terikat. Dengan enkapsulasi, data dan perilaku dapat dilindungi dari akses oleh komponen lain yang tidak perlu.  
Dalam Java, enkapsulasi dicapai dengan menggunakan kelas dan access modifier seperti private, modifier dan public. Kelas adalah unit dasar dari OOP yang mendefinisikan data dan perilaku bersama. Pengaturan akses menentukan apakah data atau perilaku dapat diakses dari kelas lain.



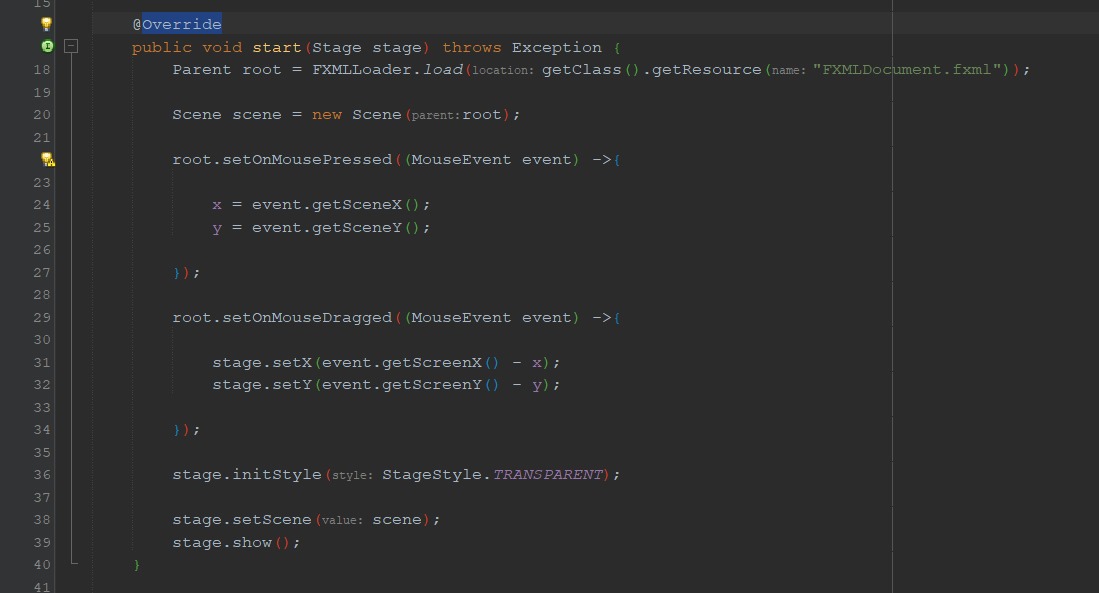
* 1. **Inheritance**

Inheritance atau pewarisan adalah konsep dalam pemrograman berorientasi objek (OOP) dimana sebuah class dapat menurunkan property dan method yang dimilikinya kepada class lain. Konsep inheritance dipakai untuk memanfaatkan fitur code reuse, yakni menghindari terjadinya duplikasi kode program. Inheritance ini juga disebut sebagai pewarisan atau penurunan.



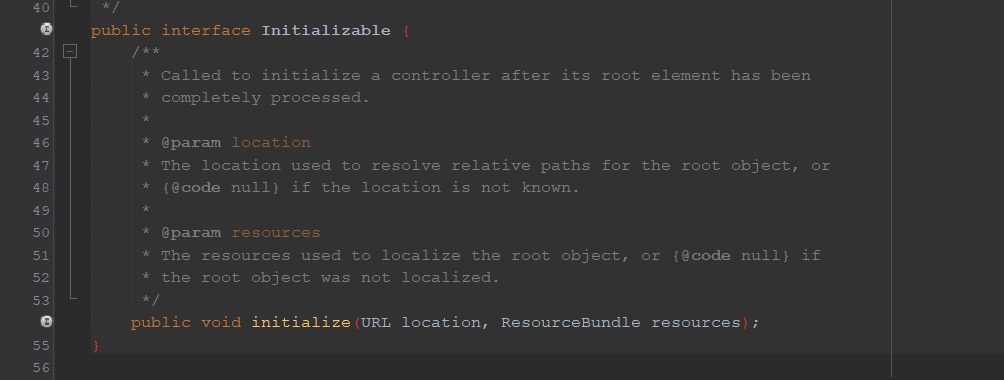
* 1. **Polymorphism**

Pada gambar dibawah ini merupakan penerapan Polymorphism dengan penggunaan anotation *@Override* yang digunakan untuk mengindikasikan bahwa method yang dideklarasikan di subclass merupakan override dari method yang ada di superclass yang merupakan sebuah konsep OOP dimana method memiliki banyak “bentuk” method yang berbeda meskipun nama yang sama



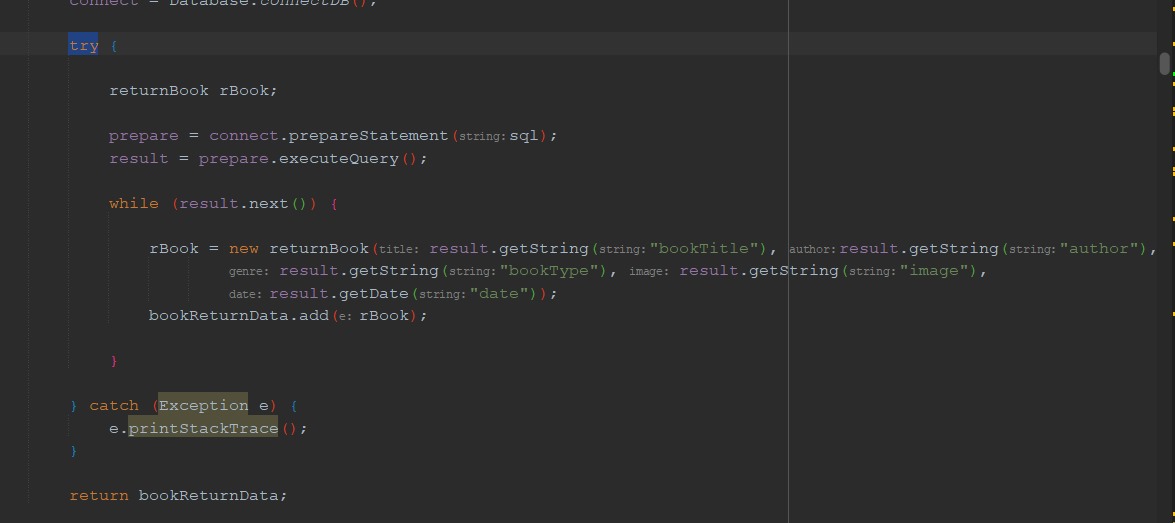
* 1. **Interface**

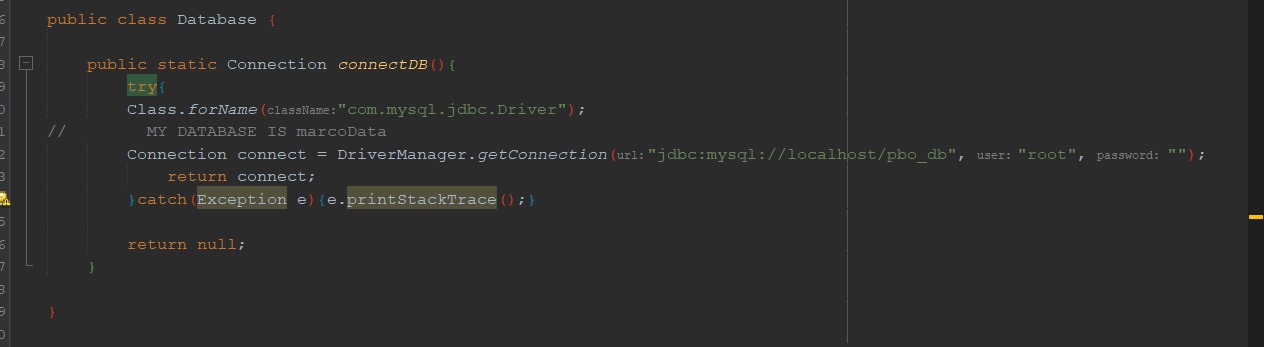
Interface merujuk pada sebuah konsep yang memungkinkan definisi dari suatu kelompok metode tanpa implementasi konkret dari metode tersebut. Dalam interface terdapat satu interface yang memiliki dua method atau lebih yang harus diimplementasikan oleh class yang menggunakannya.



* 1. **Exception handling**

Exception handling adalah konsep yang memungkinkan penanganan kesalahan yang dapat terjadi selama eksekusi program. Jika terjadi suatu kesalahan atau situasi tak terduga saat program dijalankan, *exception handling* akan memberikan cara untuk mendeteksi dan menangani situasi tersebut. *Exception Handling* biasanya diimplementasikan melalui blok try, catch, dan finally.





1. **LINK YOUTUBE**