**LAPORAN PRAKTIKUM CODELAB PEMROGRAMAN LANJUT**

**MODUL 2**

****

**Nama:** Muhammad Ibrahim Al Ayubi

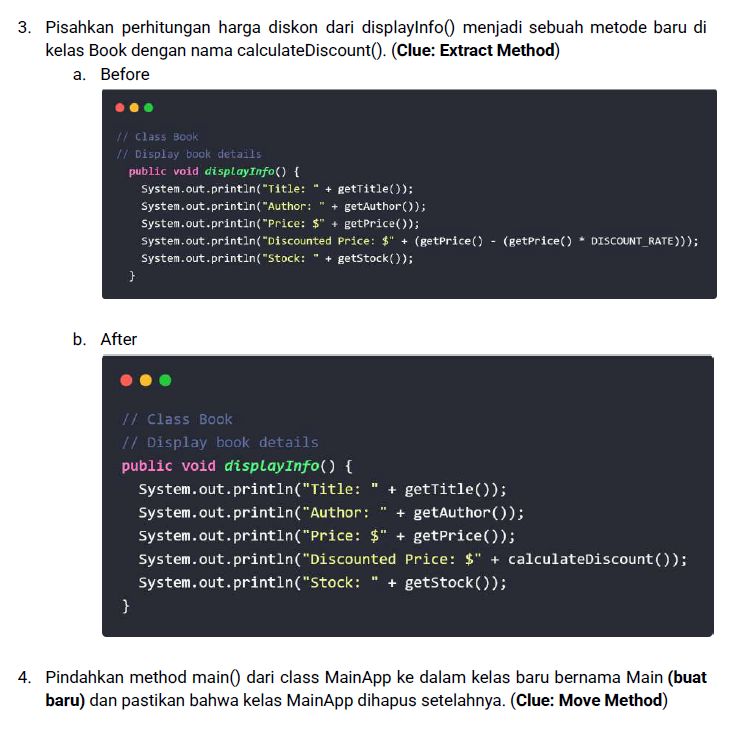
**NIM:** 202410370110123

**Kelas:** Pemrograman Lanjut E

**CODELAB:**

****

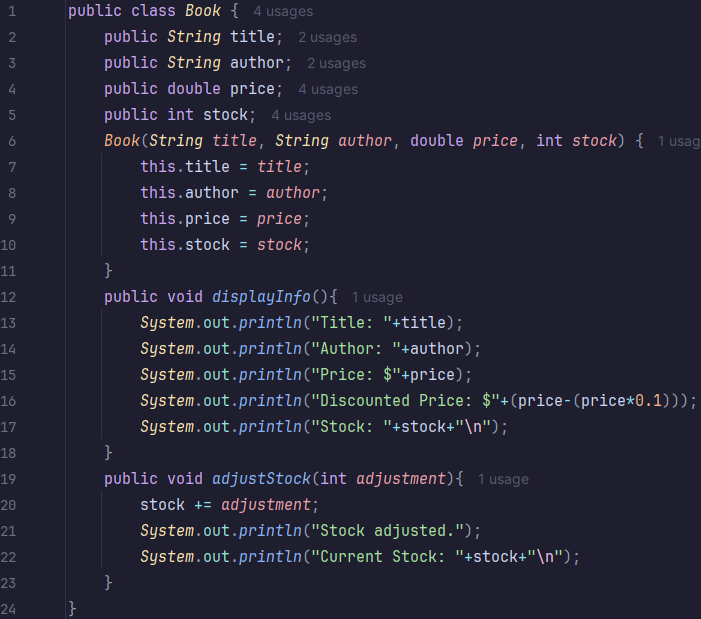
****

****

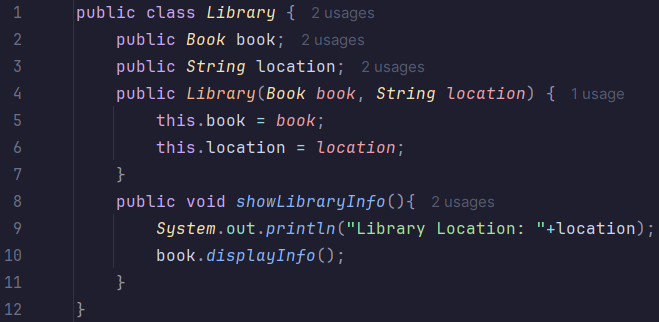
**LANGKAH-LANGKAH:**

1. Salin kode ke intellij IDE

Class Book

****

Class Library

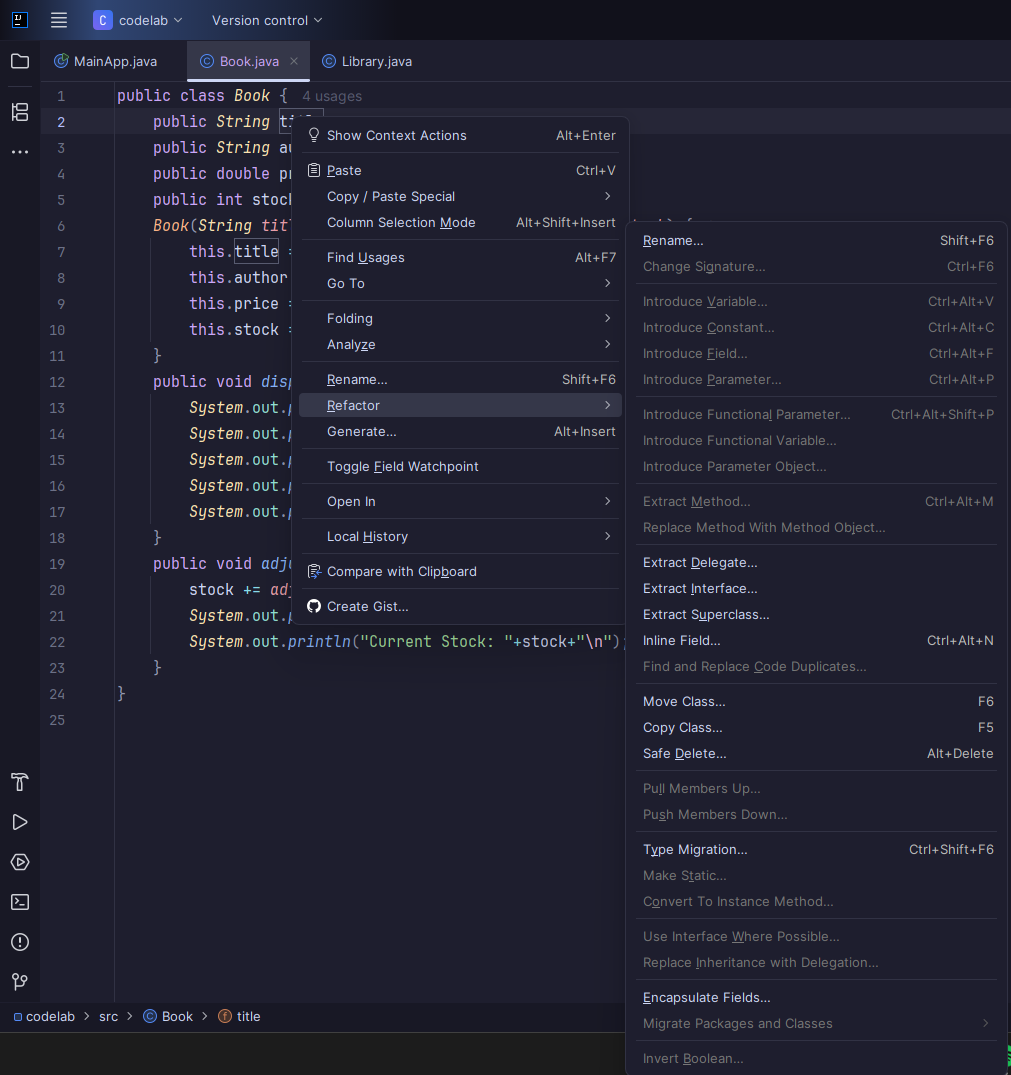


Class MainApp

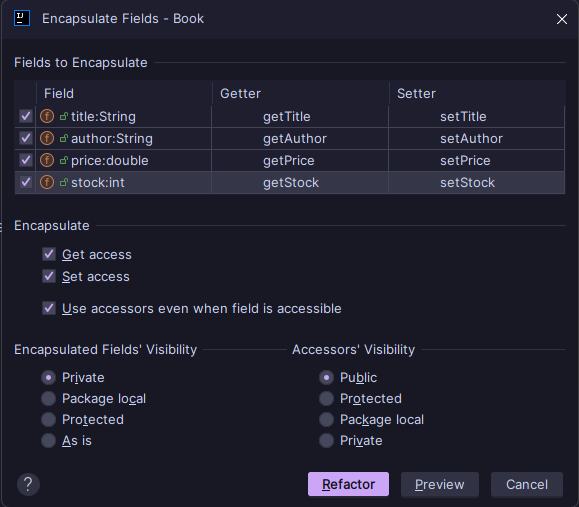


1. Selanjutnya kita akan melakukan encapsulate field di Class Book dan Class Library

* Class Book
  + Klik kanan disalah satu attribut pilih Refactor lalu klik Encapsulate field



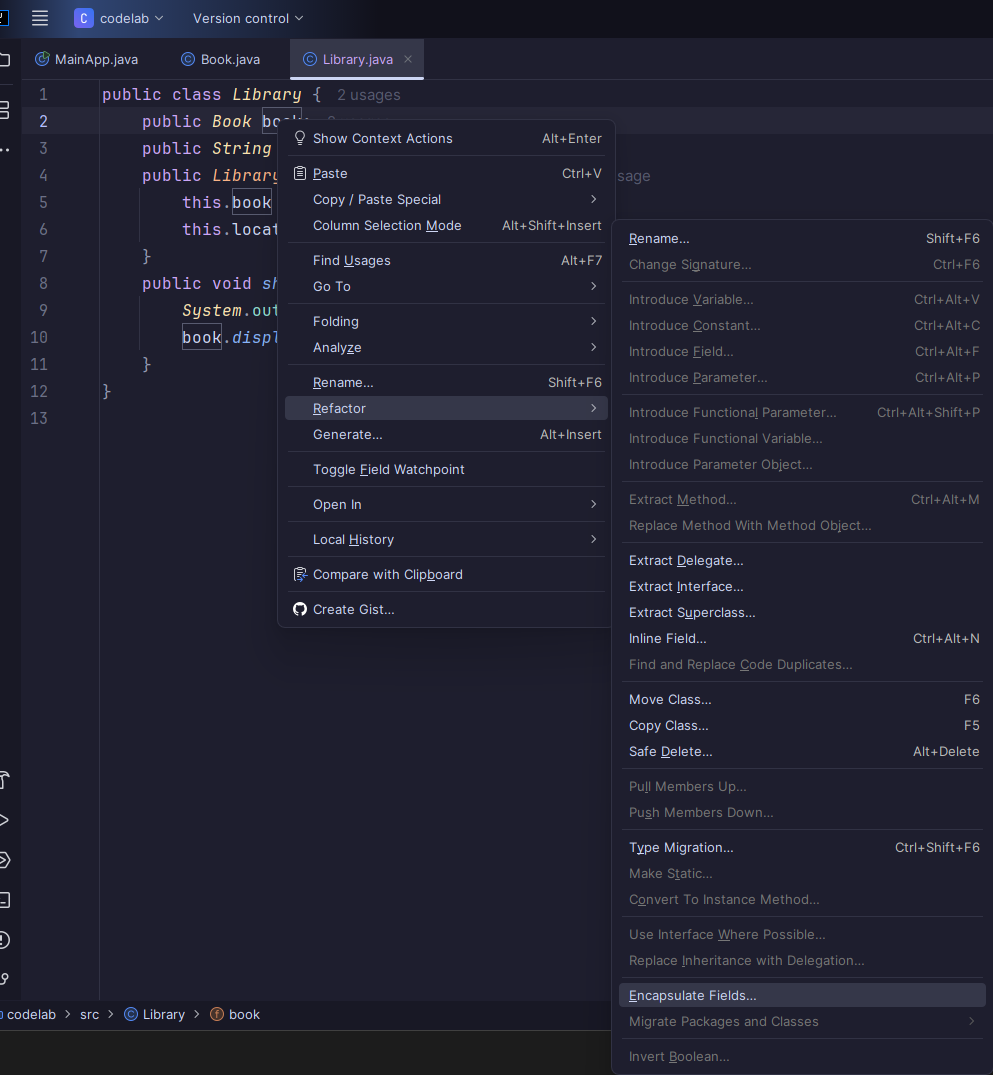
* + Ceklist field yang mau di encapsulate lalu klik refactor



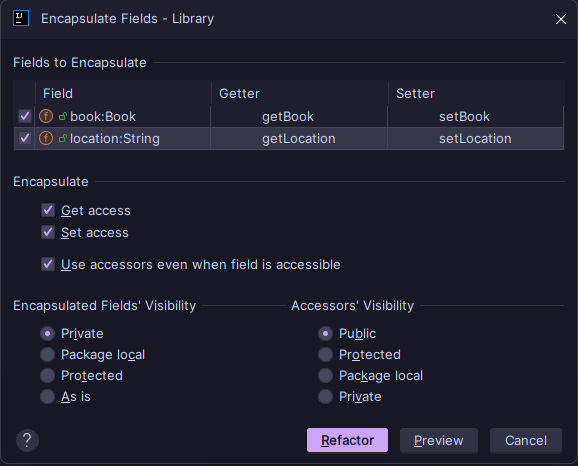
* + Maka kode akan berubah menjadi seperti ini



* Class Library
  + Klik kanan disalah satu attribut pilih Refactor lalu klik Encapsulate field



* + Ceklist field yang mau di encapsulate lalu klik refactor

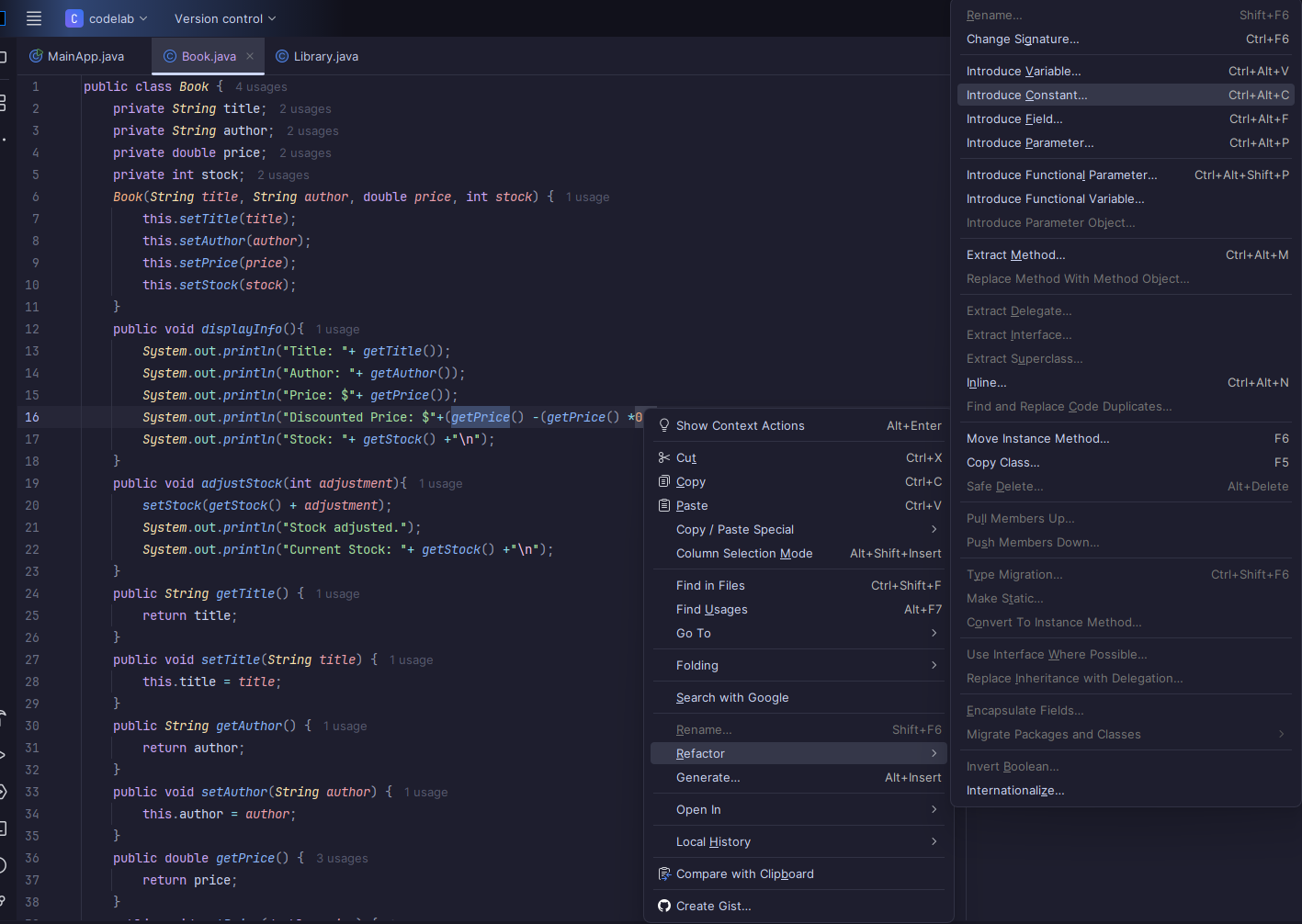


* + Maka kode akan berubah menjadi seperti ini

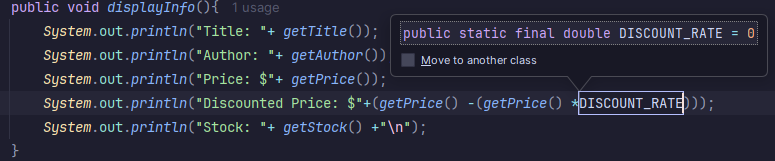


1. Lalu kita akan mengubah perhitungan diskon yang ada di class buku dengan menggunakan Introduce Constant untuk menyimpan nilai diskon

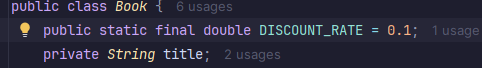
* Blok angka diskon (0.1) untuk menghitung diskon > klik kanan > refactor > introduce constant



* Ubah menjadi DISCOUNT\_RATE

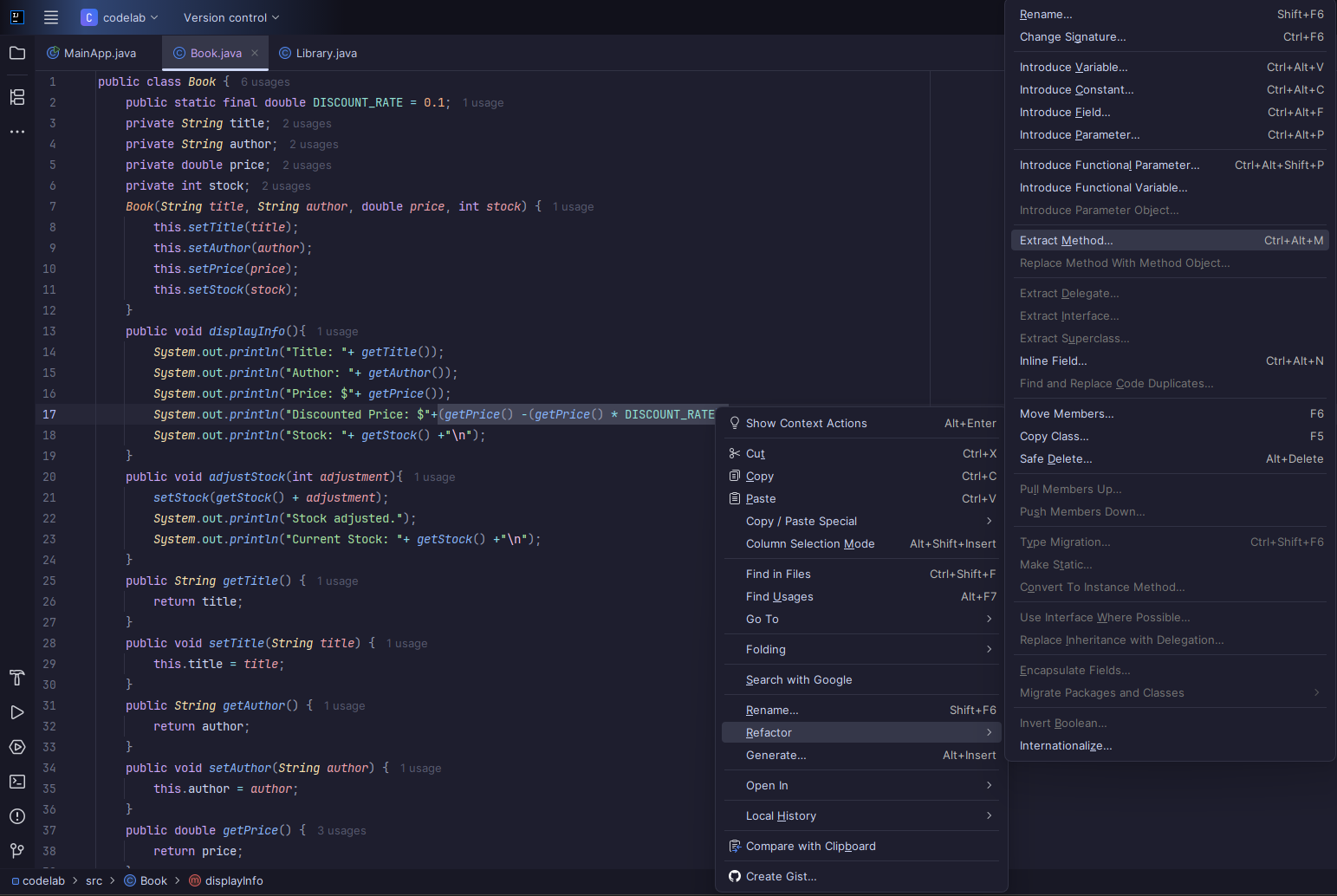


* Secara otomatis akan terbuat variable constant



1. Lalu kita akan memisahkan perhitungan harga diskon dari displayInfo () menjadi metode baru di kelas book

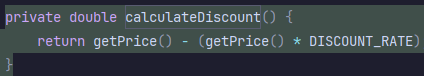
* Blok perhitungan diskon > klik kanan > refactor > Extract Method



* Ubah menjadi calculateDiscount

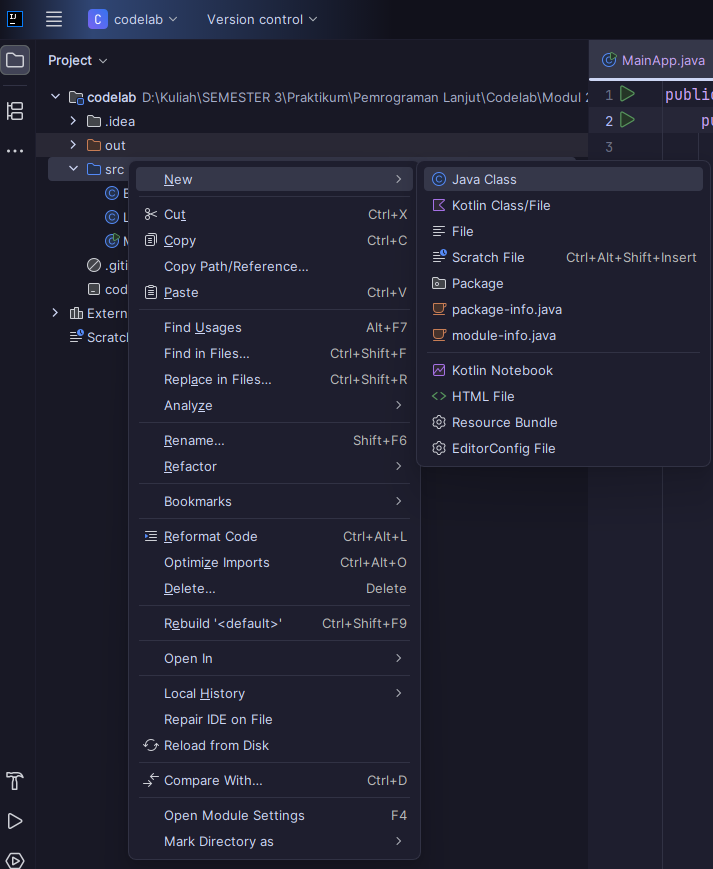


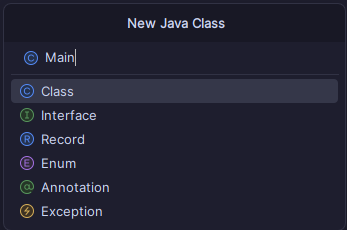
* Secara otomatis akan membuat method baru



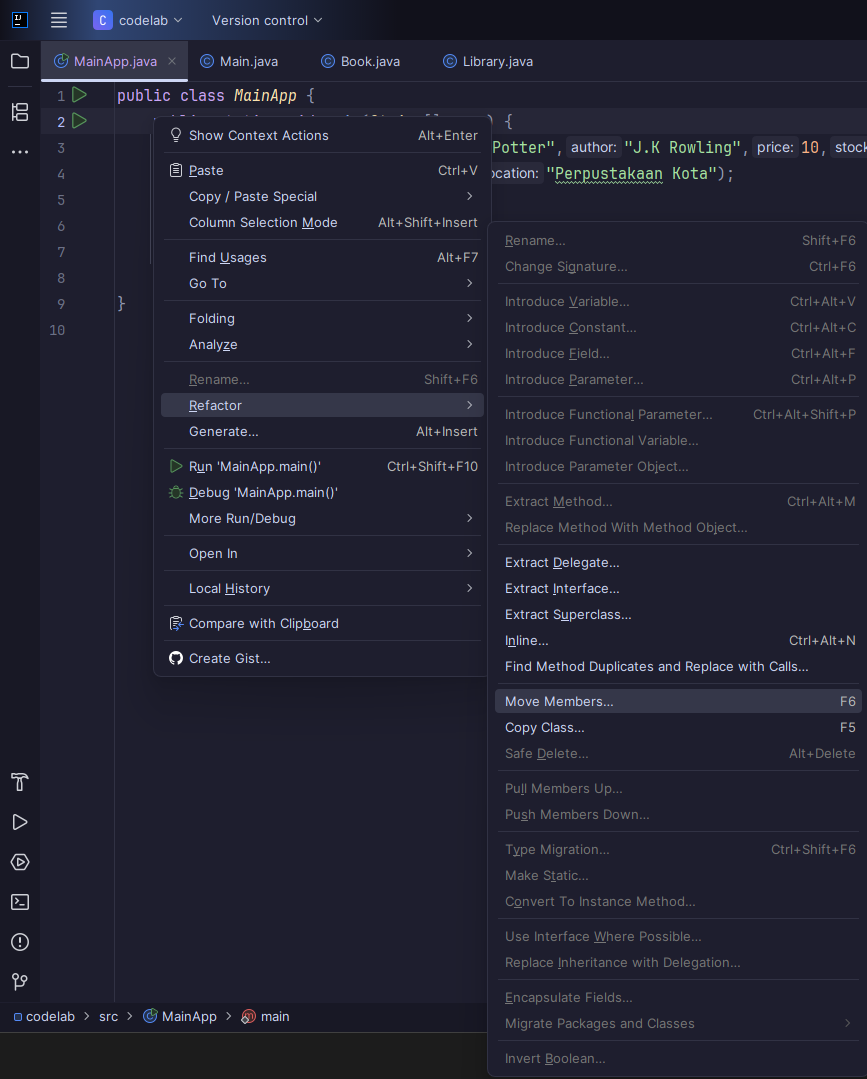
1. Terakhir kita akan memindahkan method main () dari class MainApp ke dalam kelas baru Main ()

* Buat kelas main terlebih dahulu

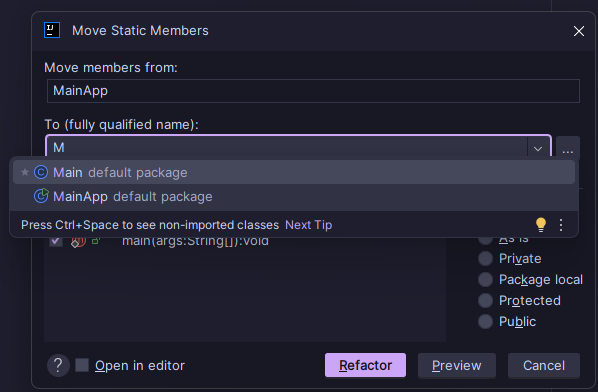


****

* Lalu balik lagi ke class MainApp () klik kanan method main > refactor > move members

****

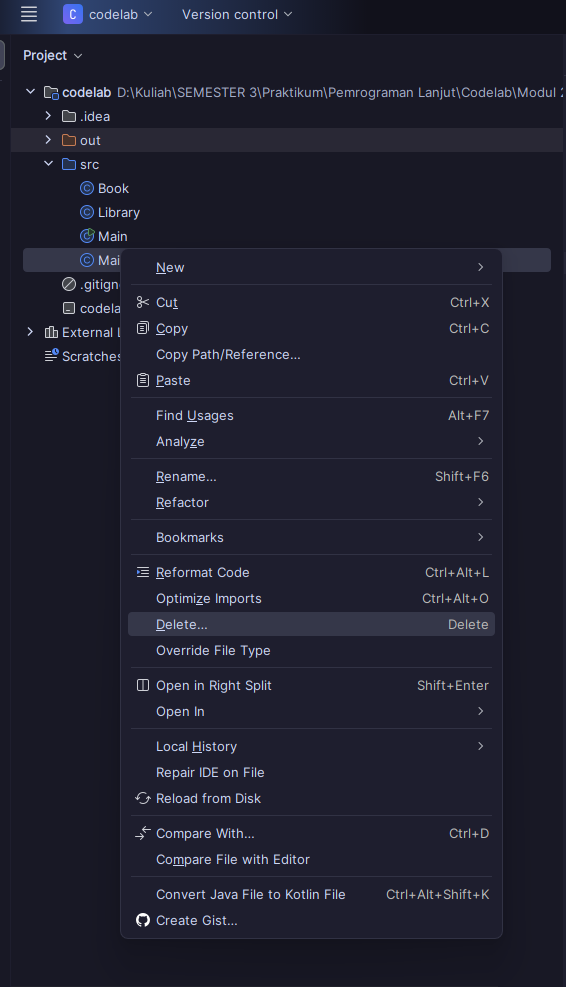
* Ketik nama Class tujuan “Main” lalu klik refactor

****

* Maka secara otomatis akan pindah ke class Main

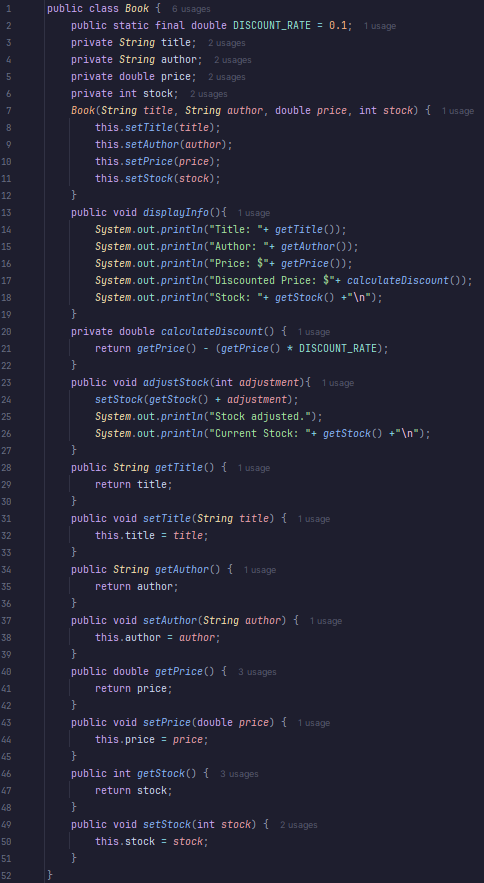
****

* Hapus Class MainApp

****

1. Maka setelah melakukang refactoring maka kode akan terlihat seperti ini

Class Book

****

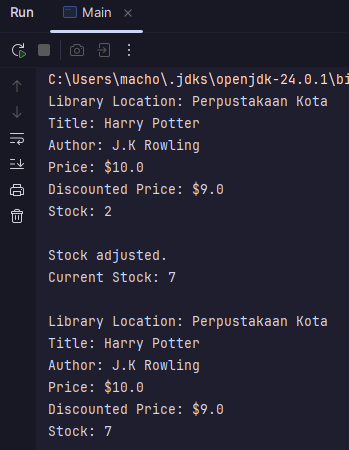
Class Library



Class Main



Output



Kesimpulan

Pada praktikum ini, kegiatan refactoring dilakukan untuk meningkatkan struktur dan keterbacaan kode program tanpa mengubah perilaku fungsionalnya. Proses refactoring diterapkan pada kode sederhana yang terdiri dari tiga kelas utama: Book, Library, dan MainApp.

Langkah-langkah refactoring yang dilakukan meliputi:

1. Encapsulation Field

Penerapan encapsulation pada kelas Book dan Library untuk melindungi data (atribut) dari akses langsung, serta meningkatkan keamanan dan modularitas kode.

1. Introduce Constant

Penggunaan konstanta DISCOUNT\_RATE menggantikan nilai diskon numerik (magic number 0.1) agar kode lebih mudah dipelihara dan dimengerti.

1. Extract Method

Pemisahan logika perhitungan diskon dari metode displayInfo() menjadi metode baru calculateDiscount() untuk meningkatkan keteraturan dan prinsip single responsibility.

1. Move Method / Move Members

Pemindahan fungsi main() dari MainApp ke kelas baru Main, sehingga struktur proyek menjadi lebih bersih dan terorganisasi.

Secara keseluruhan, hasil refactoring menghasilkan kode yang:

* Lebih terstruktur dan mudah dibaca.
* Memenuhi prinsip OOP (Object-Oriented Programming) seperti Encapsulation dan Modularity.
* Lebih mudah dikelola, diuji, dan dikembangkan di masa mendatang.