**LAPORAN HASIL PRAKTIKUM 3**

**PEMROGRAMAN BERORIENTASI OBJEK LANJUT**

**“ (Spring-2) ”**

**Dosen Pengampu :**

**Sri Hartati Wijono, M.Kom.**



Oleh

Nama : Maria Gresia Plena Br Purba

NIM : 235314094

Kelas : DP

**PROGRAM STUDI INFORMATIKA**

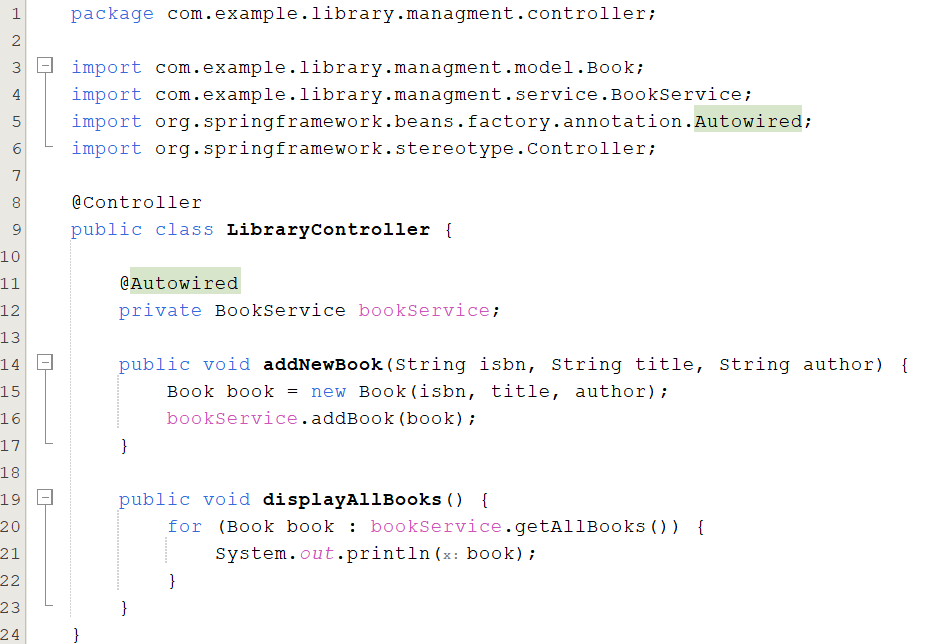
**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA**

**YOGYAKARTA**

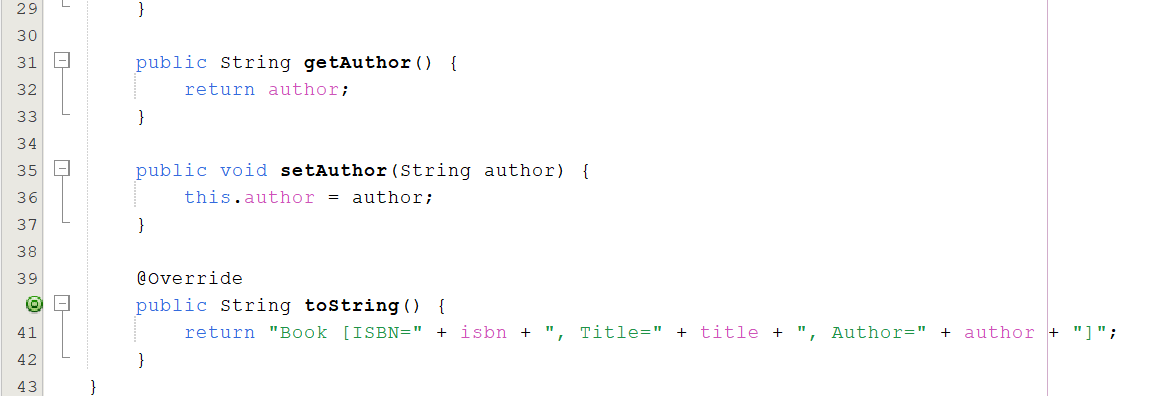
**2024**

1. **TUJUAN**
   1. Memahami penerapan **IoC** dan **DI** dalam skenario yang lebih kompleks.
   2. Mengimplementasikan berbagai jenis **Injection** (Constructor Injection, Setter Injection, Field Injection).
   3. Memahami dan menerapkan **Scopes (Singleton vs Prototype)**.
   4. Mengelola lifecycle bean menggunakan **init** dan **destroy methods**.
   5. Menggunakan anotasi **@Component**, **@Autowired**, dan **@Qualifier** untuk **Dependency Injection (DI)**.
   6. Menggunakan **Stereotype Annotations**: **@Controller**, **@Service**, dan **@Repository**
2. **PELAKSANAAN PRAKTIKUM**
   1. **Capture code class LibraryController**

****

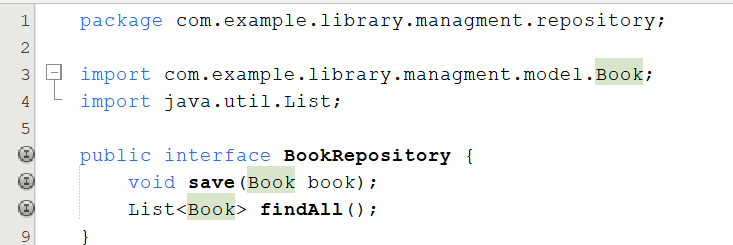
**Analisis :** pada class LibraryControllerterdapat import Autowired dan Controller. Method addNewBook() berfungsi untuk membuat objek pada class Book dengan nama book yang memiliki parameter berisi isbn, title, author. Setelah objek Book dibuat, method ini akan memanggil method addBook() dengan parameter book yang akan disimpan ke dalam objek bookService untuk menambahkan buku baru. Method displayBooks() berfungsi untuk mengembalikan dann menampilkan daftar buku. Pada method ini, terdapat pemanggilan bookService.getAllBooks() yang berfungsi untuk mendapatkan buku-buku yang ada. Lalu, akan melakukan perulangan untuk mencetak informasi dari buku.

* 1. **Capture code class Book**

****

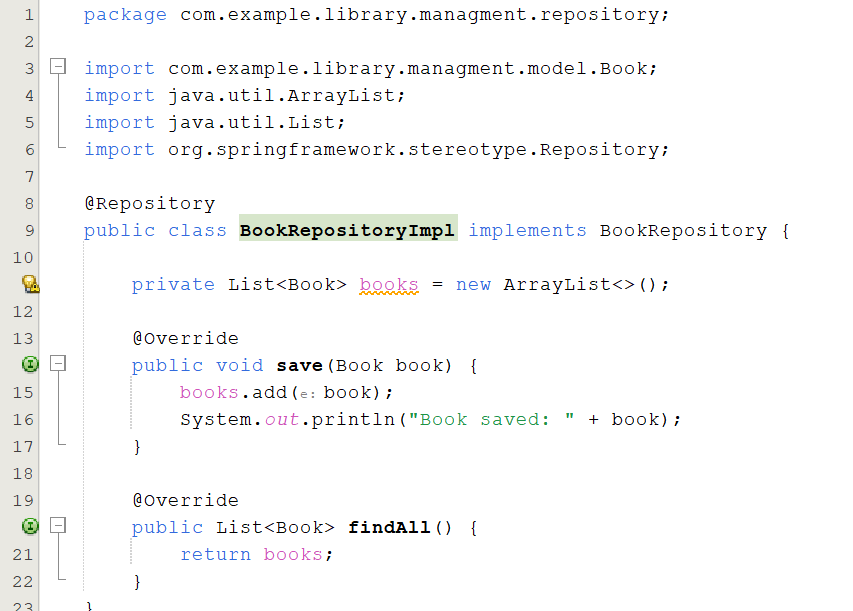
**Analisis :** terdapat beberapa method dengan setter dan getter. Getter berfungsi untuk mengembalikan nilai dari variabel-variabel, sedangkan setter untuk mengubah nilai dari variabel. Kemudian, terdapat method toString() yang akan mengembalikan informasi dari buku berupa isbn, title, dan author yang telah disimpan ke dalam variabel yang ada.

* 1. **Capture code BookRepository**

****

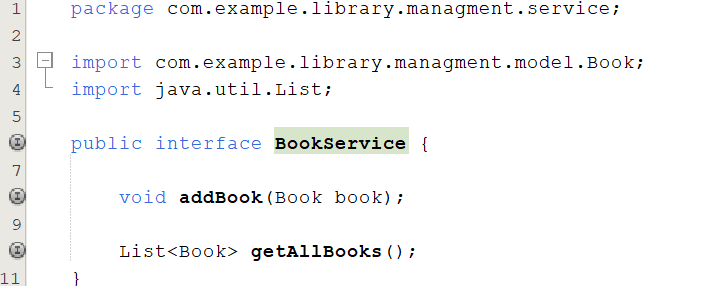
**Analisis :** interface ini memiliki method save() dengan isian parameter book yang bertipe Book. Method ini berfungsi untuk menyimpan ataupun menambahkan buku baru. Lalu, terdapat method findAll() dalam bentuk List<Book>yang berfungsi untuk mengambil dan mengembalikan buku-buku yang ada dalam bentuk List<Book>.

* 1. **Capture code class BookRepositoryImpl**

****

**Analisis :** class ini mengimplementasikan interface BookRepository. Pada class ini terdapat variabel books yang bertipe List<books> yang digunakan untuk menyimpan daftar buku dalam bentuk implementasi ArrayList<>. Kemudian, terdapat method save() dengan parameter book yang bertipe Book. Method ini akan memanggil method add pada objek books dengan parameter berisi book dan mencetak pesan jika buku tersimpan (“Book saved”). Method findAll() akan mengambil dan mengembalikan buku-buku yang ada setelah disimpan ke dalam objek Book.

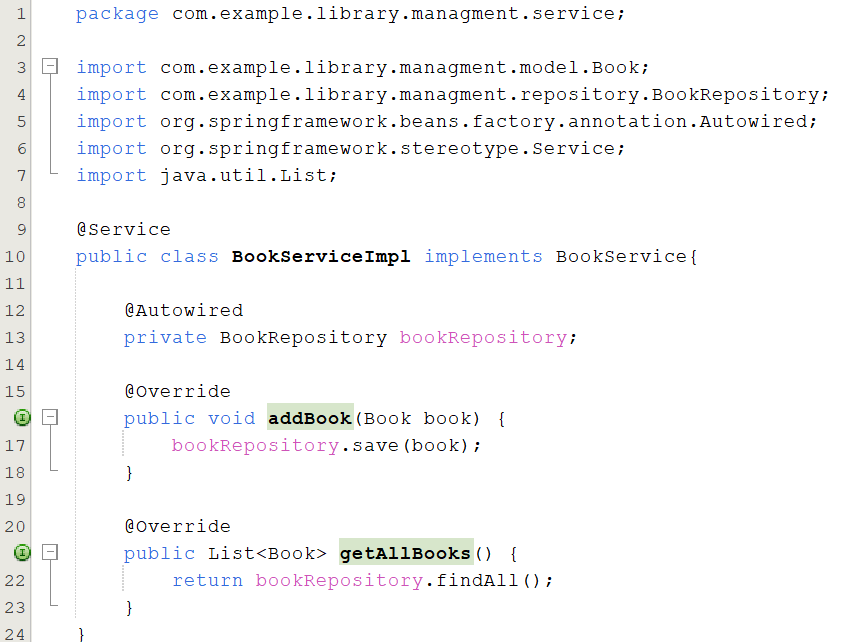
* 1. **Capture code BookService**

****

**Analisis :** interface ini memiliki method addBook() dengan parameter book yang bertipe Book. Method ini berfungsi untuk menambahkan objek Book baru. Method getAllBooks() akan mengambil dan mengembalikan buku-buku yang ada setelah disimpan ke dalam objek Book dalam bentuk List<Book>.

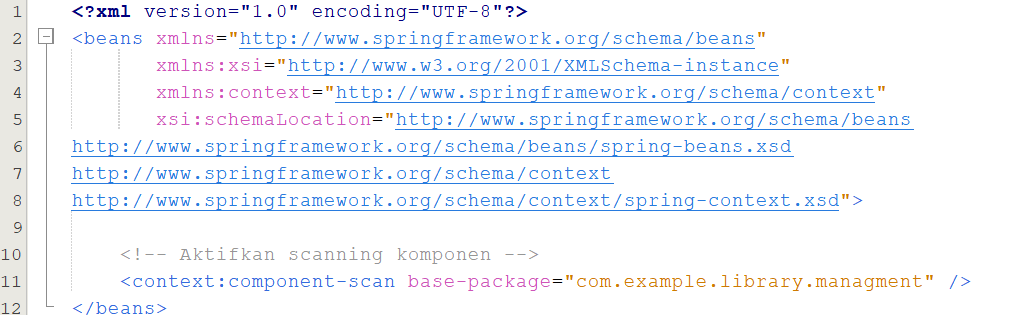
.

* 1. **Capture code class BookServiceImpl**

****

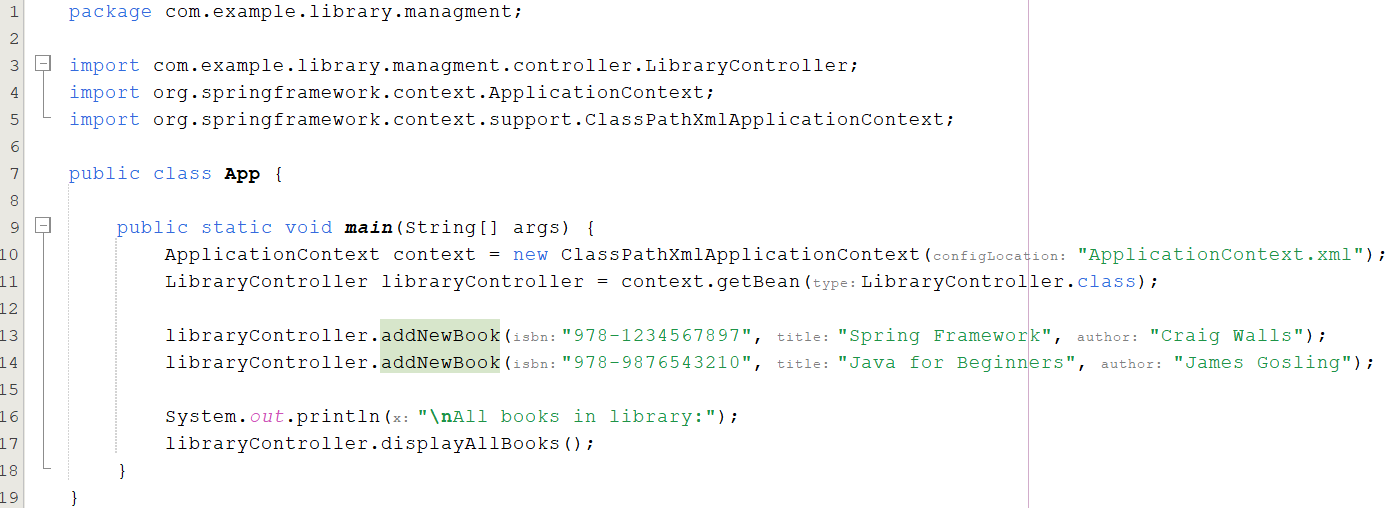
**Analisis :** class ini mengimplementasikan interface BookService. Mendeklarasikan variabel bookRepository dengan tipe BookRepository yang bersifat private. Method addBook() berfungsi untuk menambahkan buku baru. Buku yang ditambahkan akan disimpan menggunakan method save() dari bookRepository. Method getAllBooks() memanggil findAll() yang berfungsi untuk mengambil semua buku dan mengembalikannya dalam bentuk List<Book>.

* 1. **ApplicationContext.xml**

****

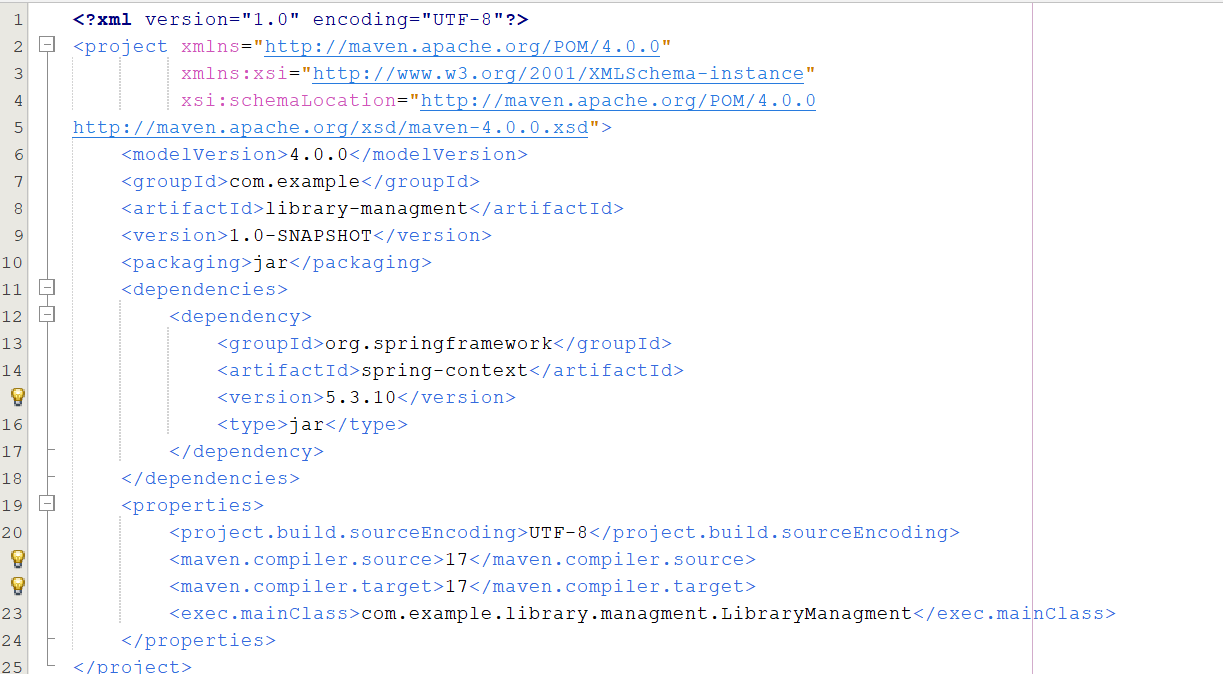
**Analisis :** xml ini merupakan konfigurasi dari Spring yang menggunakan context:component-scan untuk merujuk kepada package “com.example-library.managment”. Sehingga dapat mendaftarkan secara otomatis semua bean yang ada.

* 1. **Capture code class App**

****

**Analisis :** class ini mengimplementasikan penggunaan dari ApplicationContext yang digunakan untuk mengkonfigurasi “ApplicationContext.xml” dengan menggunakan ClassPathXmlApplicationContext. Lalu, menggunakan getBean() untuk mengambil bean LibraryController. Memanggil method addNewBook() pada objek libraryController dan mengisi data pada variabel yang ada di parameternya untuk menambahkan buku baru. Kemudian, menampilkan pesan dan memanggil method displayAllBooks() pada objek libraryController untuk menampilkan semua buku yang ada.

* 1. **Pom.xml**

****

**Analisis :** dependencies berfungsi untuk mendeklarasikan pustaka eksternal, yaitu groupId, artifactId, dan version. Lalu, pada properties terdapat encoding, versi dari java yang digunakan, dan mainClass yang akan dijalankan. Pom ini berfungsi untuk memudahkan pengelolaan proyek.

* 1. **Capture Output :**

