**1. Code Smell: Duplicate Code**

* **File: Semua servlet (AddBookServlet.java, CartServlet.java, ViewBookServlet.java, dll)**
* **Masalah: Banyak kode yang berulang di setiap servlet seperti validasi login, pengaturan content type, dan penanganan error.**
* **Solusi: Membuat kelas BaseServlet yang mengimplementasikan Template Method pattern untuk menstandarisasi alur proses servlet dan menghilangkan duplikasi kode.**

**2. Code Smell: Long Method**

* **File: AddBookServlet.java, CartServlet.java, ViewBookServlet.java**
* **Masalah: Method service() terlalu panjang dan melakukan terlalu banyak tanggung jawab.**
* **Solusi: Memecah method menjadi beberapa method yang lebih kecil dan fokus seperti doBusinessLogic(), handleNotLoggedIn(), dan processAddBook().**

**3. Code Smell: Poor Exception Handling**

* **File: BookServiceImpl.java, UserServiceImpl.java**
* **Masalah: Blok catch kosong atau hanya memanggil e.printStackTrace() tanpa penanganan yang tepat.**
* **Solusi: Menambahkan logging yang proper dan melempar exception yang lebih spesifik dengan pesan yang jelas.**

**4. Code Smell: Resource Leak**

* **File: BookServiceImpl.java, UserServiceImpl.java**
* **Masalah: Resource database (Connection, PreparedStatement, ResultSet) tidak ditutup dengan benar.**
* **Solusi: Menambahkan method closeResources() di DBUtil dan menggunakannya di blok finally.**

**5. Code Smell: Primitive Obsession**

* **File: AddBookServlet.java, RemoveBookServlet.java**
* **Masalah: Penggunaan isBlank() yang tidak kompatibel dengan versi Java yang lebih lama.**
* **Solusi: Mengganti dengan trim().isEmpty() untuk memastikan kompatibilitas.**

**6. Code Smell: Feature Envy**

* **File: StoreUtil.java**
* **Masalah: Method updateCartItems() terlalu kompleks dan melakukan terlalu banyak operasi.**
* **Solusi: Memecah method menjadi addToCart() dan removeFromCart() yang lebih fokus.**

**7. Code Smell: SQL Injection Vulnerability**

* **File: BookServiceImpl.java**
* **Masalah: Method getBooksByCommaSeperatedBookIds() menggunakan konkatenasi string untuk query SQL.**
* **Solusi: Mengimplementasikan prepared statement dengan parameter untuk mencegah SQL injection.**

**8. Code Smell: Lack of Logging**

* **File: Semua file**
* **Masalah: Tidak ada logging yang memadai untuk debugging dan monitoring.**
* **Solusi: Menambahkan logger dan pesan log yang informatif di seluruh aplikasi.**

**9. Code Smell: Syntax Error**

* **File: CustomerRegisterServlet.java**
* **Masalah: Kesalahan sintaks pada BookStoreCons tants.CONTENT\_TYPE\_TEXT\_HTML.**
* **Solusi: Memperbaiki menjadi BookStoreConstants.CONTENT\_TYPE\_TEXT\_HTML.**

**10. Code Smell: Inappropriate Intimacy**

* **File: Servlet classes**
* **Masalah: Servlet memiliki akses langsung ke atribut sesi yang seharusnya dikelola oleh kelas lain.**
* **Solusi: Menambahkan method getSessionAttribute() dan setSessionAttribute() di BaseServlet.**

**11. Code Smell: Lack of Cohesion**

* **File: ViewBookServlet.java, CartServlet.java**
* **Masalah: Kelas melakukan terlalu banyak hal yang tidak terkait.**
* **Solusi: Memisahkan tanggung jawab dengan memindahkan logika bisnis ke method terpisah.**

**12. Code Smell: Magic Numbers/Strings**

* **File: Semua servlet**
* **Masalah: Penggunaan string literal untuk pesan error dan nilai konstanta.**
* **Solusi: Menggunakan logger untuk pesan error dan konstanta untuk nilai yang sering digunakan.**

**Refactoring ini mengikuti prinsip-prinsip dari Martin Fowler dan Girish Suryanarayana, menjadikan kode lebih mudah dipelihara, diuji, dan lebih robust.**