**1. Code Smell: Long Method (Bloaters)**

* **File:** AddBookServlet.java
* **Bagian:** service method
* **Masalah:** Metode service terlalu panjang dan melakukan banyak tugas, seperti validasi, pengolahan data, dan rendering tampilan. Ini membuat kode sulit dibaca dan dipelihara.
* **Solusi:** Pisahkan logika menjadi beberapa metode yang lebih kecil dan spesifik, seperti metode untuk validasi, pengolahan data, dan rendering tampilan.

**2. Code Smell: Duplicated Code (Bloaters)**

* **File:** AboutServlet.java, ErrorHandlerServlet.java
* **Bagian:** Penggunaan RequestDispatcher dan StoreUtil.setActiveTab
* **Masalah:** Kode untuk memeriksa login dan mengatur tab aktif diulang di beberapa servlet.
* **Solusi:** Ekstrak kode yang berulang ke dalam metode utilitas atau superclass yang dapat digunakan kembali oleh semua servlet.

**3. Code Smell: Primitive Obsession (Bloaters)**

* **File:** CustomerRegisterServlet.java
* **Bagian:** Penggunaan tipe data primitif untuk phone dan email
* **Masalah:** Menggunakan tipe data primitif untuk data yang memiliki aturan validasi khusus, seperti nomor telepon dan email.
* **Solusi:** Gunakan objek nilai (value objects) untuk mewakili data ini, yang dapat mengenkapsulasi logika validasi.

**4. Code Smell: Large Class (Bloaters)**

* **File:** CartServlet.java
* **Bagian:** Seluruh kelas
* **Masalah:** Kelas ini memiliki terlalu banyak tanggung jawab, termasuk pengelolaan sesi, pengolahan data, dan rendering tampilan.
* **Solusi:** Pisahkan tanggung jawab ke dalam kelas atau komponen yang lebih kecil dan lebih fokus.

**5. Code Smell: Feature Envy (Object-Orientation Abusers)**

* **File:** ProcessPaymentServlet.java
* **Bagian:** Metode service
* **Masalah:** Metode ini terlalu bergantung pada detail implementasi dari BookService dan HttpSession.
* **Solusi:** Pindahkan logika yang terkait dengan BookService dan HttpSession ke dalam kelas atau metode yang lebih sesuai.

Setelah mengidentifikasi *code smell* ini, langkah berikutnya adalah melakukan refactoring untuk menghilangkan atau mengurangi *code smell* tersebut. Saya akan mulai dengan refactoring kode Anda berdasarkan analisis ini.

**Refactoring Plan**

1. **Extract Method**: Break down long methods into smaller, more manageable methods.
2. **Extract Class**: For large classes, split responsibilities into separate classes.
3. **Introduce Utility Classes**: For duplicated code, create utility classes or methods.
4. **Use Value Objects**: Replace primitive types with objects that encapsulate validation logic.
5. **Encapsulate Field**: For fields that are accessed directly, provide getter and setter methods.

**Refactoring Steps**

**Step 1: Refactor AddBookServlet.java**

* **Extract Method**: Break down the service method into smaller methods like validateBookDetails, processAddBook, and renderAddBookForm.

**Step 2: Refactor AboutServlet.java and ErrorHandlerServlet.java**

* **Introduce Utility Class**: Create a utility class for common operations like checking login status and setting active tabs.

**Step 3: Refactor CustomerRegisterServlet.java**

* **Use Value Objects**: Create Email and PhoneNumber classes to encapsulate validation logic.

**Step 4: Refactor CartServlet.java**

* **Extract Class**: Separate session management, data processing, and rendering into different classes.

**Step 5: Refactor ProcessPaymentServlet.java**

* **Encapsulate Field**: Use methods to interact with HttpSession and BookService.