

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam proses pengembangan perangkat lunak, penyusunan kode yang rapi dan bersih merupakan tantangan yang sering dihadapi pengembang. Tekanan untuk menyelesaikan proyek dalam waktu singkat sering menyebabkan pengembang mengabaikan praktik penulisan kode yang baik. Akibatnya, muncul kode yang rumit, sulit dipahami, tidak terstruktur, serta rentan terhadap kesalahan. Kondisi ini menimbulkan kesulitan dalam proses pemeliharaan, terutama ketika aplikasi harus diperbarui atau disesuaikan dengan kebutuhan baru. Kode yang tidak terorganisir juga menghambat kolaborasi tim karena pengembang membutuhkan waktu lebih lama untuk memahami logika yang tidak konsisten, sehingga menurunkan efisiensi dan produktivitas kerja.

Prinsip *clean code* menekankan bahwa kode harus berfungsi dengan baik sekaligus mudah dipahami oleh berbagai pengembang yang terlibat dalam proyek. Namun, dalam praktiknya, masih sering ditemukan kode dengan struktur yang tidak rapi, penamaan variabel yang tidak jelas, serta komentar berlebihan yang justru mengganggu keterbacaan dan pemeliharaan. Permasalahan tersebut dikenal dengan istilah *code smells*, yaitu tanda adanya kelemahan struktural dalam kode yang dapat menghambat pengembangan sistem lebih lanjut[1].

Penelitian ini berfokus pada penerapan prinsip *clean code* dalam pengembangan aplikasi manajemen *Salon Bliss* berbasis web menggunakan framework Laravel. Penerapan prinsip ini menghasilkan kode yang terstruktur, jelas, mudah dipahami, dan efisien sehingga mendukung pengembangan jangka panjang. Aplikasi dirancang untuk membantu *Salon Bliss* dalam mengelola layanan, pelanggan, jadwal pemesanan, karyawan, serta transaksi secara daring dan waktu nyata. Pendekatan ini mengurangi masalah seperti duplikasi kode, penamaan fungsi yang tidak konsisten, serta bagian kode yang tidak diperlukan. Sistem informasi ini meningkatkan efisiensi operasional salon dan memberikan layanan yang cepat, tepat, serta akurat kepada pelanggan[2].

Penelitian ini menggunakan metode *smells* dan *heuristics* untuk mengevaluasi kualitas kode. *Code smells* berfungsi mengidentifikasi kelemahan seperti penamaan yang tidak deskriptif dan logika yang kompleks, sedangkan *heuristics* menjadi panduan untuk pengambilan keputusan desain kode yang lebih baik. Kombinasi kedua metode ini memungkinkan pengembang menemukan masalah secara sistematis dan menerapkan perbaikan yang tepat[3].

Refactoring dilakukan untuk memperbaiki kode yang mengandung masalah. Melalui penerapan prinsip clean code seperti penghapusan duplikasi dan penamaan fungsi yang jelas, aplikasi *Salon Bliss* menjadi lebih optimal, modular, serta mudah dikembangkan. *Refactoring* juga meningkatkan fleksibilitas sistem dalam penambahan fitur baru tanpa menurunkan stabilitas dan performa aplikasi[4].

Penerapan *clean code* tidak hanya meningkatkan kualitas teknis kode, tetapi juga memperlancar operasional sistem. Dalam aplikasi *Salon Bliss*, struktur kode yang terorganisir memungkinkan pengelolaan data secara akurat dan efisien, serta mempermudah adaptasi sistem terhadap kebutuhan baru seperti integrasi fitur laporan pemesanan. Dengan demikian, aplikasi menjadi responsif terhadap perubahan kebutuhan bisnis salon yang dinamis[5].

Kode yang mengandung *code smells* seperti struktur berulang, dokumentasi berlebihan, atau nama variabel yang tidak jelas dapat memperlambat proses pemeliharaan dan memicu kesalahan yang sulit dilacak. Kondisi ini menimbulkan kendala dalam kolaborasi tim, karena pengembang membutuhkan waktu tambahan untuk memahami struktur kode yang tidak efisien. Penerapan prinsip *clean code* menjadi solusi untuk meminimalkan masalah tersebut sehingga kode menjadi lebih mudah dipahami dan meningkatkan efektivitas kerja tim[6].

Secara keseluruhan, penelitian ini menegaskan pentingnya penerapan prinsip *clean code* dalam pengembangan aplikasi manajemen *Salon Bliss*. Aplikasi ini mendukung pengelolaan pelanggan, penjadwalan layanan, pencatatan transaksi, serta pelaporan pendapatan secara terintegrasi. Melalui analisis *code smells*, *heuristics*, dan *Refactoring*, kualitas kode dapat ditingkatkan sehingga menjadi lebih terorganisir, mudah dipahami, dan berkelanjutan. Hasil akhirnya adalah aplikasi manajemen salon yang efisien, terstruktur, dan mampu mendukung strategi

bisnis *Salon Bliss* dalam meningkatkan kualitas layanan serta daya saing di industri kecantikan[7].

Meskipun penelitian mengenai penerapan *clean code* dan *Refactoring* telah banyak dilakukan, sebagian besar penelitian hanya berfokus pada penggunaan tools otomatis untuk mendeteksi *code smells* tanpa melakukan analisis manual yang sistematis berdasarkan heuristik *clean code*. Selain itu, belum banyak penelitian yang mengintegrasikan deteksi manual dan otomatis serta mengevaluasi perbandingan kualitas kode sebelum dan sesudah *Refactoring* secara kuantitatif. Oleh karena itu, diperlukan penelitian yang tidak hanya melakukan perbaikan kode, tetapi juga menyusun tahapan deteksi manual yang terstruktur dan mengevaluasi dampaknya terhadap peningkatan kualitas perangkat lunak.

1.2 Rumusan Masalah

Bagaimana Menerapkan prinsip *clean code* menggunakan metode *smells* dan *heuristics* dalam proses *Refactoring* aplikasi *Salon Bliss* dapat meningkatkan keterbacaan, efisiensi, serta kemudahan pemeliharaan kode?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah untuk meningkatkan keterbacaan, efisiensi, dan kemudahan pemeliharaan kode aplikasi *Salon Bliss* melalui penerapan prinsip *clean code* dengan pendekatan *smells* dan *heuristics* dalam proses *Refactoring*.

1.4 Manfaat

Adapun manfaat dari penelitian ini ialah:

1. Penelitian ini berkontribusi dalam pengembangan perangkat lunak melalui proses *Refactoring* dengan menerapkan prinsip *clean code* menggunakan pendekatan *code smells* dan *heuristics*, guna meningkatkan efisiensi pengembangan dan kemudahan pemeliharaan kode pada aplikasi *Salon Bliss*.
2. Membantu pengembang memahami penerapan *heuristics Clean Code* dan *refactoring* dalam memperbaiki *code smells* sehingga struktur kode menjadi lebih rapi, mudah dipahami, dan lebih terstruktur.

3. Menghasilkan struktur kode yang lebih modular sehingga memudahkan proses pemeliharaan, pengembangan, dan kolaborasi antar pengembang aplikasi.

1.5 Batasan Masalah

1. Penelitian ini hanya dilakukan pada aplikasi *Salon Bliss* berbasis Laravel, dengan fokus analisis pada bagian *controller* dan *service*.
2. Identifikasi code smell dilakukan secara manual berdasarkan prinsip *Clean Code* dan pendekatan smells serta heuristics, tanpa membahas analisis menggunakan *tools* otomatis.
3. Penelitian ini hanya menganalisis jenis *code smell* yang ditemukan pada proses observasi dan tidak membahas seluruh kemungkinan jenis *code smell* yang ada dalam teori.
4. Validasi hasil *refactoring* dilakukan bersama dua orang programmer berdasarkan perbandingan struktur kode sebelum dan sesudah dilakukan perbaikan.