

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya era digital ini, perkembangan sistem perangkat lunak dituntut untuk memiliki kualitas yang baik, fleksibel dalam melakukan pengembangan, serta mudah dalam melakukan pemeliharaan. Namun, pada kenyataannya, masih banyak pengembang atau developer perangkat lunak yang menghasilkan kode program yang tidak terstruktur, sulit dibaca, serta menyulitkan proses pengembangan lebih lanjut. Hal ini disebabkan oleh kurangnya pemahaman dan penerapan prinsip desain yang tepat dalam proses pengembangan perangkat lunak.

Salah satu solusi untuk mengatasi permasalahan tersebut adalah dengan menerapkan prinsip desain SOLID. Prinsip SOLID merupakan seperangkat pemodan dalam pemograman berorientasi objek. Maka dengan menerapkan prinsip SOLID ini pengembang dapat membangun struktur kode yang bersih, terorganisir, dan mudah untuk diuji. Sehingga mampu meningkatkan efisiensi pemeliharaan serta kualitas perangkat lunak secara keseluruhan.[1]

SOLID *Principle* pertama kali diperkenalkan oleh *Robert C. Martin (Uncle Bob)* sebagai panduan desain dalam pemrograman berorientasi objek. SOLID Principle ini memiliki lima prinsip utama yaitu *Single Responsibility*, *Open/Closed*, *Liskov Subsitution*, *Interface Segregation*, dan *Depedency Inversion* bertujuan untuk membat asrsitekture perangkat lunak yang lebih tangguh terhadap perubahan, serta memudahkan kalaborasi dalam melakukan pengembangan. Penerapan prinsip ini terbukti mampu meningkatkan keterbacaan, pengujian, dan pemeliharaan perangkat lunak, terutama pada project bersekala menengah hingga besar.

Prinsip desain SOLID telah banyak digunakan untuk pengembangan perangkat lunak modren untuk menciptakan sistem yang modular dan fleksibel. Pendekatan ini memberikan panduan dalam penyusunan kode yang lebih terstruktur, sehingga memudahkan proses perawatan dan dan pengembangan lebih

lanjut. Dengan menerapkan prinsip SOLID, perubahan pada sistem dapat diakomodasi tanpa menimbulkan ketergantungan kompleks antar komponen.[2]

Penelitian ini dilakukan dengan studi kasus pada aplikasi *Happy Thrifting*, yaitu sebuah platform berbasis web untuk jual beli baju bekas secara online. Pada sistem yang ada sebelumnya, struktur kode yang digunakan belum tertata dengan baik, belum modular, dan masih sulit dipahami. Pada skripsi ini dilakukan pengembangan ulang struktur kode dengan menerapkan prinsip desain SOLID, sehingga menghasilkan kode yang lebih bersih, modular, dan mudah diuji. Penerapan ini juga meningkatkan keteraturan struktur kode, mempermudah proses pemeliharaan, serta mendukung pengembangan sistem secara berkelanjutan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah dalam penelitian ini difokuskan pada bagaimana proses penerapan prinsip desain SOLID dalam pengembangan aplikasi *web Happy Thrifting* agar dapat menghasilkan struktur sistem yang modular dan terorganisir. Selain itu, penelitian ini juga bertujuan untuk mengetahui cara membangun arsitektur perangkat lunak yang mematuhi prinsip-prinsip tersebut, serta bagaimana mengevaluasi efektivitas penerapan prinsip SOLID tersebut secara kuantitatif dengan menggunakan perangkat analisis *PHP Metrics*.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan prinsip SOLID dalam pengembangan aplikasi web *Happy Thrifting* untuk menciptakan sistem yang lebih modular dan *maintainable*.
2. Merancang struktur kode yang terorganisir dan fleksibel dengan berpedoman pada prinsip SOLID.
3. Mengevaluasi hasil penerapan SOLID dari sisi kemudahan pemeliharaan menggunakan metode *static code analysis* dengan bantuan *tools PHP Metrics*.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Membuat aplikasi *Happy Thrifting* lebih modular dan mudah dipelihara untuk kedepannya.

2. Memberikan referensi praktis bagi pengembang perangkat lunak dalam menerapkan prinsip desain SOLID dalam pengembangan aplikasi web.
3. Menambah wawasan akademik terkait pentingnya struktur kode yang baik dan modular dalam pengembangan perangkat lunak.