

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Kualitas kode merupakan salah satu faktor yang perlu diperhatikan pada saat mengembangkan suatu perangkat lunak. Menurut riset, kualitas kode dalam suatu perangkat lunak sangatlah berpengaruh terhadap waktu rilis suatu perangkat lunak maupun kualitas perangkat lunak yang akan dihasilkan. Dalam sebuah penelitian setelah dianalisis menggunakan *Static Code Analysis Tool* (Codescene) suatu kode program dengan kategori “*Alert*” menghasilkan waktu pengembangan 124% kali lebih lama dari kode program yang memiliki kategori “*Health*”, sedangkan untuk kategori “*Warning*” suatu perangkat lunak menghasilkan waktu pengembangan 78% lebih lama dari kode program yang memiliki kategori “*Health*” [1].

*Website Symposium* merupakan sebuah *website* yang diperuntukan untuk para pembicara suatu seminar/konferensi dalam mengelola informasi yang berhubungan dengan pekerjaan mereka sebagai pembicara seperti topik pembicaraan, biodata, jadwal konverensi hingga pencatatan status pengajuan proposal CFP. *Website Symposium* dapat digunakan pembicara dalam mencari lowongan pekerjaan pembicara sedangkan penyelenggara seminar/konferensi dapat memanfaatkan *website* ini sebagai tempat untuk mencari tenaga kerja seorang pembicara.

Berdasarkan hasil wawancara kepada salah satu pengembang *website Symposium*, diketahui bahwa beberapa module kode pada *website* ini sulit untuk dilakukan pengembangan oleh pengembang terutama bagian yang menjadi fitur utama pada *website* ini seperti module *Conferences*, *Talk* dan *Account* karena kompleksitas modulnya yang cukup tinggi **LAMPIRAN A**. Dengan mengacu pada hasil wawancara, peneliti memutuskan untuk melakukan pengujian *maintainability* menggunakan *tools PhpMetrics* untuk memastikan masalah *maintainabilitynya* dan hasilnya ternyata banyak *class* yang memiliki nilai *Maintainability Index* yang termasuk kedalam kalasifikasi “*Low Maintainability*”

yaitu memiliki nilai dibawah 64. Observasi secara langsung terhadap kode sumber juga dilakukan dan ditemukan adanya beberapa masalah seperti *function* yang menjalankan banyak tugas sekaligus, adanya duplikasi algoritma, tidak adanya abstraksi data, hingga adanya penamaan yang kurang *explicit* **LAMPIRAN B.**

Dengan adanya permasalahan yang ada maka peneliti menyimpulkan bahwa perlu adanya solusi pada sistem yang sedang berjalan saat ini agar sistem yang ada dapat lebih mudah untuk di kembangkan kedepannya. Solusi yang digunakan oleh peneliti untuk memecahkan masalah yang ada adalah dengan mengimplementasikan metode *Clean Code*. *Clean Code* merupakan suatu metode yang digunakan untuk membuat kode sumber suatu perangkat lunak dapat lebih mudah untuk dipahami dan dimaintain [2]. Dengan menerapkan *Clean Code* diharapkan kode sumber dari *website* Symposium dapat lebih mudah untuk *maintenance* oleh pengembang *website* ini kedepannya.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah disampaikan diatas, maka dapat disimpulkan rumusan masalahnya adalah apakah dengan menerapkan metode *Clean Code* dapat meningkatkan *maintainability* dari kode sumber *website* Symposium.

## **1.3 Maksud Dan Tujuan**

Maksud dari penelitian ini adalah menerapkan *Clean Code* dan *Design Pattern* pada *website* Symposium. Adapun tujuan dari maksud tersebut adalah untuk mengoptimasi *maintainability* kode sumber *website* Symposium, agar *website* ini menjadi lebih mudah untuk dilakukan pengembangan kedepannya.

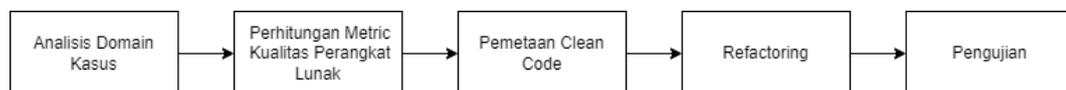
## **1.4 Batasan Masalah**

Adapun ada beberapa batasan masalah yang diterapkan pada penelitian ini guna menjaga konsistensi dari penelitian ini, berikut merupakan batasan yang diterapkan;

1. Penelitian hanya berfokus pada bagian kode sumber yang sudah ada dan tidak mengubah/menambah fitur yang telah dibuat sebelumnya.
2. Fokus kode yang akan direfactor hanyalah pada bagian *class* jenis *Controller* saja tanpa mengubah struktur maupun bentuk code bagian *class* jenis *View* dan juga *Model*.
3. Metode yang digunakan untuk meningkatkan kualitas *maintainability* kode sumber *website* Symposium menggunakan metode *Clean Code* yang terdapat pada buku karya Robert Cecil Martin berjudul *Clean Code: A Handbook of Agile Software Craftsmanship*.

### 1.5 Metodologi Penelitian

Jenis metodologi penelitian yang digunakan pada penelitian ini adalah penelitian analisis dengan pendekatan kuantitatif. Metode penelitian analisis berfokus pada melakukan analisa terhadap suatu kasus dengan fakta yang tersedia dilanjutkan dengan melakukan evaluasi terhadap kasus yang ada. Metodologi penelitian ini terdiri dari beberapa tahap yang dapat dilihat pada gambar dibawah ini;



Gambar 1.1 Tahapan Penelitian

Berikut merupakan penjelasan tahapan pada metodologi penelitian ini;

#### 1. Analisis Domain Kasus

Pada tahap ini peneliti akan menganalisis dan menjelaskan lebih rinci mengenai domain *website* Symposium.

#### 2. Perhitungan Metrik Kualitas Perangkat Lunak

Pada tahap ini kode akan dilakukan perhitungan metrik kualitas untuk mengklaksifikasikan bagian area kode mana saja yang perlu dipetakan masalahnya.

#### 3. Pemetaan Clean Code

Pada tahap ini kode sumber *website* Symposium akan dipetakan berdasarkan buku *Clean Code*.

#### **4. Refactoring**

Pada tahap ini kode yang telah dipetakan akan diperbaiki dengan menggunakan teknik *refactoring* tanpa mengubah external-behaviournya guna meningkatkan kualitas kode program.

#### **5. Pengujian**

Pada tahap ini kode akan dilakukan pengujian dan dibuktikan apakah kode yang telah *direfactor* terbukti dapat meningkatkan *maintainability* dari *website* Symposium.

### **1.6 Sistematika Penulisan**

Sistematika penulisan penelitian ini disusun guna memberikan gambaran umum mengenai penelitian yang akan dilakukan. Penulisan pada penelitian kali ini akan dibagi menjadi 5 bab utama, yaitu sebagai berikut;

#### **BAB 1 PENDAHULUAN**

Pada BAB 1 akan membahas mengenai Latar Belakang, Rumusan Masalah, Maksud dan Tujuan, Batasan Masalah, serta Metodologi Penelitian dan Sistematika Penelitian.

#### **BAB 2 LANDASAN TEORI**

Pada BAB 2 akan membahas mengenai teori-teori yang akan digunakan dalam mendukung penulisan penelitian kali ini.

#### **BAB 3 ANALISIS DAN PERANCANGAN**

Pada BAB 3 akan membahas mengenai isi dari penelitian kali ini yaitu mulai dari tahap analisis domain kasus hingga pemetaan masalah kode program.

#### **BAB 4 IMPLEMENTASI DAN PENGUJIAN**

Pada BAB 4 akan membahas mengenai proses *refactoring* dan juga pengujian terhadap kode yang telah dilakukan *refactoring*.

#### **BAB 5 KESIMPULAN DAN SARAN**

Pada BAB 5 akan membahas mengenai kesimpulan yang didapatkan pada penelitian kali ini serta berisi saran-saran yang berguna untuk pengembangan penelitian ini lebih lanjut.