

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam prariset yang telah dilakukan dengan melaksanakan inspeksi terhadap bagian *back end* dari aplikasi *content marketing* yang dibangun oleh Divisi Codelabs dan wawancara terhadap pengembang terkait. Ditemukan permasalahan berupa tidak konsistennya struktur modul, terdapat duplikasi dari kode sumber, dan penamaan yang tidak terlalu eksplisit baik pada nama berkas maupun variabel pada aplikasi. Selain itu, setelah melakukan wawancara kepada dua pengembang baru dari aplikasi yang diteliti, ditemukan pula bahwa temuan-temuan sebelumnya mempersulit proses pemahaman kode sumber dari aplikasi. Apabila temuan-temuan tersebut dibiarkan, dapat membuat *maintainability* pada aplikasi tersebut rendah, sehingga berkemungkinan besar mengakibatkan proses pemeliharaan dan pengembangan selanjutnya menjadi terhambat. Adapun bukti dari hasil inspeksi kode sumber dapat dilihat pada **Error! Reference source not found..**

Berdasarkan fenomena tersebut, dilakukan studi literatur mengenai kualitas perangkat lunak. Ditemukan bahwa dalam menilai suatu perangkat lunak menurut McCall dilihat dari 3 perspektif, yaitu *Product Revision*, *Product Transition*, dan *Product Operation* di mana setiap perspektif memiliki faktor penilaiannya masing-masing [1]. Pada penelitian ini, perangkat lunak yang diteliti dinilai dari perspektif *Product Revision* yang menilai kemampuan suatu perangkat lunak dalam menghadapi perubahan setelah perangkat lunak tersebut sudah melewati tahap pengembangan, di mana faktor yang diambil dari perspektif tersebut adalah faktor *maintainability*.

Secara umum, faktor *maintainability* menilai seberapa mudah suatu aplikasi untuk dipelihara (*maintain*) [2], meliputi dokumentasi dari perangkat lunak baik dari internal program maupun dari segi kode sumber. Oleh karena itu, baik penulisan kode sumber dan dokumentasi dari perangkat lunak perlu diperhatikan agar proses pemeliharaan dapat dengan mudah dilakukan.

Dalam menulis kode, terdapat sebuah konsep yang bernama *Clean Code*. *Clean code* merupakan suatu konsep yang menerangkan bahwa kode yang baik adalah kode yang mudah untuk dipahami, baik dari penamaan maupun struktur dari kode itu sendiri [3]. Dalam penggunaannya, *clean code* sendiri menyarankan pemanfaatan *design pattern* dalam membangun suatu modul agar lebih terstruktur. *design pattern* merupakan sebuah konsep untuk menyelesaikan masalah yang biasa terjadi dalam desain perangkat lunak dengan solusi yang sudah ada (*Reusable solution*) [4].

Dengan demikian, dalam upaya meningkatkan *maintainability* pada aplikasi *content marketing* akan diterapkan konsep *clean code* dan *design pattern* pada aplikasi tersebut. *clean code* digunakan untuk memperbaiki tingkat keterbacaan dari kode sumber yang ada agar dapat mudah dipahami oleh pengembang. Sedangkan *design pattern* digunakan untuk memperbaiki struktur modul agar lebih mudah untuk dipelihara.

1.2 Perumusan Masalah

Berdasarkan pemaparan dari latar belakang, maka dapat diidentifikasi masalah pada penelitian ini adalah apakah dengan menerapkan *clean code* dan *design pattern* dapat meningkatkan *maintainability* pada aplikasi *content marketing*.

1.3 Maksud dan Tujuan

Maksud dilaksanakannya penelitian ini adalah melakukan implementasi *design pattern* dan konsep *clean code*. Adapun tujuan dari penelitian ini, yaitu meningkatkan *maintainability* pada aplikasi yang diteliti.

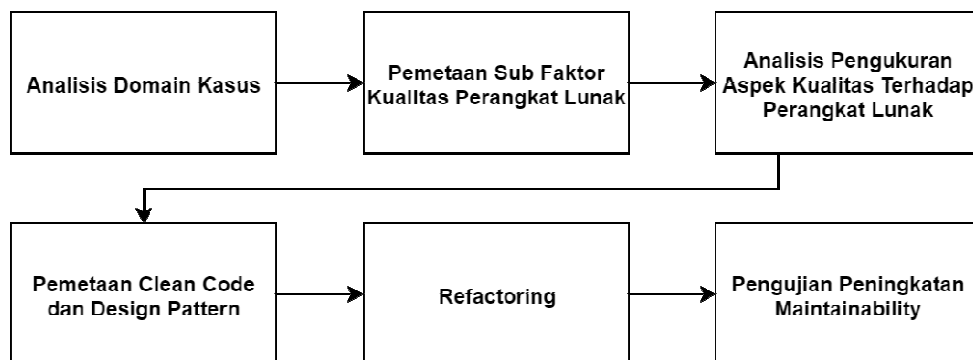
1.4 Batasan Masalah

Dalam penelitian ini, masalah yang diangkat dibatasi agar tidak menyimpang dari tujuan yang ingin dicapai. Adapun batasan masalah pada penelitian ini adalah:

- a. Analisis dan paradigma pemrograman menggunakan pendekatan berorientasi objek.
- b. Penilaian yang dilakukan hanya meliputi kode dan dokumentasi dari aplikasi.
- c. Implementasi *clean code* dan *design pattern* menggunakan *Test-Driven Development*.
- d. *Design pattern* yang digunakan mengikuti petunjuk yang terdapat pada buku *Learning Javascript Design Pattern* (Addy Osmani, 2012).
- e. Keluaran atau artefak dari penelitian ini berupa dokumentasi dari perangkat lunak yang terdiri atas *code convention* dan dokumentasi fungsional dari aplikasi yang diteliti.

1.5 Metodologi Penelitian

Jenis penelitian yang dilakukan merupakan penelitian kuantitatif [5]. Metodologi pada penelitian ini mengacu pada buku *Refactoring to Patterns* [6]. Metodologi tersebut selanjutnya disesuaikan dengan kebutuhan penelitian. Langkah-langkah yang dilakukan pada penelitian ini dapat dilihat pada Gambar 1.1.



Gambar 1.1 Tahapan Penelitian

Berikut merupakan penjelasan dari tahapan penelitian yang terdapat pada penelitian ini:

a. Analisis Domain Kasus

Pada tahapan ini dilakukan analisis terhadap aplikasi berupa analisis sistem secara umum meliputi analisis kebutuhan fungsional dengan menggunakan pendekatan berorientasi objek dan analisis masalah yang ditemukan pada aplikasi.

b. Pemetaan Sub Faktor Kualitas Perangkat Lunak

Dalam tahapan ini dilakukan pemetaan terhadap sub faktor kualitas perangkat lunak yang akan dianalisis.

c. Analisis Pengukuran Aspek Kualitas Terhadap Perangkat Lunak

Pada tahap ini dilakukan pengukuran terhadap perangkat lunak dengan menilai modul-modul yang pada perangkat lunak tersebut, dan menentukan tindakan selanjutnya terhadap modul yang dinilai.

d. Pemetaan *Clean Code* dan *Design Pattern*

Pada tahap ini dilakukan pemetaan konsep *clean code* dan *design pattern* terhadap modul yang sudah dinilai sebelumnya.

e. *Refactoring*

Pada tahap ini dilakukan *refactoring* terhadap modul yang ada pada aplikasi *content marketing*, di mana *clean code* dan *design pattern* yang sudah dipetakan diimplementasikan.

f. Pengujian Peningkatan *Maintainability*

Pada tahap ini dilakukan pengujian terhadap tingkat *maintainability* di mana perangkat lunak sudah melewati tahap implementasi.

1.6 Sistematika Penulisan

Sistematika penulisan proposal ini disusun untuk memberikan gambaran umum mengenai penelitian yang dikerjakan. Sistematika penulisan dalam proposal skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

BAB I berisi tentang latar belakang permasalahan, merumuskan inti permasalahan, menentukan maksud dan tujuan, pembatasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

BAB II berisi tentang bahan-bahan kajian, konsep dasar, dan teori dari para ahli yang berkaitan dengan penelitian. Meninjau permasalahan dan hal-hal yang berguna dari penelitian-penelitian dan menggunakannya sebagai acuan pemecahan masalah pada penelitian ini.

BAB III ANALISIS DAN PEMETAAN CLEAN CODE DAN DESIGN PATTERN

BAB III berisi tentang tahapan analisis dari subjek penelitian untuk mengetahui hal atau masalah yang timbul dari penelitian yang dilakukan dan melakukan penentuan solusi yang tepat untuk masalah yang ditemukan.

BAB IV *REFACTORING* DAN PENGUJIAN *MAINTAINABILITY*

BAB IV berisi tentang *refactoring* dan pengujian *maintainability* yang dilakukan. Pada bagian ini juga akan ditentukan bagaimana teknik penanganan diimplementasi, diuji, dan disesuaikan dengan hasil penelitian.

BAB V KESIMPULAN DAN SARAN

BAB V berisi tentang kesimpulan dari hasil penelitian beserta saran untuk pengembangan selanjutnya.

