

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini, kita butuh media yang mudah diakses yang dapat membantu pengidap HIV/AIDS untuk berkonsultasi dan berbagi dengan orang-orang yang memiliki kemampuan untuk memberikan dukungan emosional. Oleh karena itu, muncullah inovasi-inovasi untuk mengatasi masalah tersebut, salah satunya perangkat lunak yang dikembangkan oleh tempat penelitian ini. Inovasi tersebut tertuang dalam sebuah perangkat lunak pintar yang berperan sebagai media edukasi dan informasi mengenai HIV/AIDS bernama *Comrades*, yang berisi konten seperti berita, artikel dan sejumlah *event* dalam sebuah perangkat lunak android, *website* dan terintegrasi dengan *chatbot*. Berdasarkan dari wawancara kepada pengembang terkait, terdapat permasalahan dalam proses pengembangan berkenaan dengan kode sumber, khususnya tingkat keterbacaan dan struktur modul, hal ini menyulitkan proses pengembangannya dan sehingga menyulitkan proses *turn over* dari developer lama ke developer baru. Salah satu yang menjadi faktor penentu kualitas perangkat lunak dapat dilihat dari sisi bagaimana kode sumbernya dapat dirawat dan dikembangkan dengan mudah, sehingga perangkat lunak *comrades* dapat dikatakan bermasalah dalam sisi kualitas kode sumber. Berdasarkan hasil dari inspeksi kode sumber dan wawancara kepada tim pengembang perangkat lunak *comrades* yang dapat dilihat pada **LAMPIRAN C**, ditemukan permasalahan pada kode sumber khususnya di bagian API. Beberapa dari permasalahan tersebut berupa bagian-bagian pada kode sumber yang berulang, masalah kompleksitas (*nested*), bagian kode yang diberi komentar namun tidak dijelaskan dan penamaan variabel yang tidak konsisten. Hal tersebut menyulitkan dalam pemahaman dan pengembangan ke depan. Jika dibiarkan, masalah-masalah tersebut dapat memperlambat proses pengembangan dan pemeliharaan kode sumber[1]. Hasil inspeksi dari permasalahan tersebut di atas dapat dilihat pada **LAMPIRAN A**.

Dalam konsep kualitas perangkat lunak, terdapat sebuah konsep *Software Quality Assurance (SQA)*, yang terdiri dari *Code Quality*, *Requirement Quality*, *Design Quality* dan *Quality Control Effectiveness*. Berdasarkan permasalahan dari hasil wawancara, maka penelitian ini akan berfokus kepada *Code Quality* dan atribut yang dinilai adalah *Complexity*, *Maintainability* dan *Documentation*[2]–[4]. Dengan demikian, penulisan kode sumber dan dokumentasi dari perangkat lunak perlu diperhatikan agar proses pemeliharaan atau *Maintainability* dapat dengan mudah dilakukan. Berbicara mengenai penulisan kode sumber perangkat lunak, dikenal sebuah konsep yang bernama *clean code*. Konsep tersebut menjelaskan bahwa perangkat lunak yang baik, mudah dikembangkan dan dipelihara itu harus memiliki kode sumber mudah dipahami, baik dari segi konsistensi struktur dan penamaan maupun keterbacaan[5]. Konsep lain yang bernama *design pattern* juga memiliki peran penting. *Design Pattern* adalah konsep pola penulisan kode yang *reusable* untuk menyelesaikan masalah yang sering terjadi dalam sumber kode sehingga dapat menghasilkan kode sumber yang lebih berstruktur[6].

Dengan demikian, dalam upaya meningkatkan tingkat *maintainability* perangkat lunak *comrades* akan diimplementasikan konsep *clean code* dan *design pattern*. *Clean code* bertujuan untuk meningkatkan keterbacaan agar mudah dipahami dan dipelihara, sedangkan *design pattern* bertujuan untuk memperbaiki struktur modulnya.

## **1.2 Rumusan Masalah**

Apakah dengan menerapkan konsep *clean code* dan *design pattern* dapat meningkatkan kualitas perangkat lunak *comrades* dari segi *maintainability* agar lebih mudah dipelihara dan dikembangkan?

## **1.3 Maksud dan Tujuan**

Maksud dari penelitian ini adalah melakukan implementasi *clean code* dan *design pattern* pada perangkat lunak *comrades*. Sedangkan tujuan dari penelitian

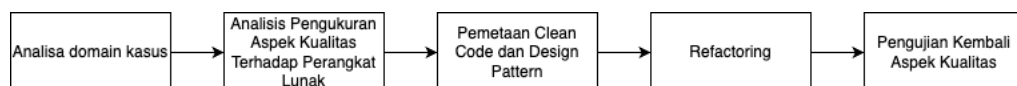
ini adalah untuk meningkatkan kualitas perangkat lunak *comrades* dari segi *maintainability* agar lebih mudah dipelihara dan dikembangkan.

#### 1.4 Batasan Masalah

- a. Penilaian yang dilakukan hanya meliputi kode dan dokumentasi dari perangkat lunak.
- b. Penelitian ini hanya berfokus kepada implementasi konsep *clean code* dan *design pattern*.
- c. *Output* dari penelitian ini berupa dokumentasi dari perangkat lunak yang meliputi *code convention* dan dokumentasi fungsional perangkat lunak.
- d. Konsep analisis yang digunakan pada penelitian ini adalah dengan pendekatan pemrograman berorientasi objek.

#### 1.5 Metodologi Penelitian

Penelitian ini menggunakan metode kuantitatif yang berlandaskan pada buku *Refactoring To Patterns*[7]. Diagram berikut secara umum menggambarkan tahapan penelitian yang akan dilakukan:



**Gambar 1.1 Tahapan Penelitian**

Penjelasan dari setiap tahap penelitian:

- a. Analisis Domain Kasus

Tahap ini akan dilakukan analisis sistem yang secara umum meliputi analisis kebutuhan fungsional dan analisis permasalahan yang ada pada perangkat lunak dengan pendekatan berorientasi objek.

- b. Analisis Pengukuran Aspek Kualitas Terhadap Perangkat Lunak.

Perangkat lunak yang akan dianalisis akan dilakukan pengukuran awal untuk mendapatkan nilai kualitasnya pada saat sebelum dilakukan penelitian yang

kemudian akan dibandingkan dengan hasil akhir. Hasil pengukuran juga akan dianalisis untuk menjadi acuan terhadap tindakan pada tahap berikutnya.

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

Hasil Analisa pada tahap sebelumnya, kemudian akan dilakukan pemetaan konsep *clean code* apa saja yang diperlukan dan diimplementasikan terhadap kode sumber perangkat lunak.

d. Refactoring.

Kode sumber yang sudah dianalisa, akan dilanjutkan ke tahap refactoring sebagai bentuk implementasi *clean code* dan *design pattern*.

e. Pengujian Kembali Tingkat *maintainability*

Pada tahap akhir ini, nilai kualitas perangkat lunak yang telah di refactor akan dinilai kembali untuk mengamati apakah tahapan-tahapan implementasi *clean code* terbukti atau tidaknya dapat meningkatkan kualitas perangkat lunak dari sisi *maintainability*.

## **1.6 Sistematika Penulisan**

Berikut akan dipaparkan bagaimana struktur penulisan proposal untuk memberi gambaran umum terhadap penelitian yang akan dilakukan pada laporan proposal skripsi ini:

### **BAB I PENDAHULUAN**

BAB I membahas mengenai latar belakang, permasalahan, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

BAB II berisi mengenai bahasan mengenai hasil, kajian, pendapat para ahli, konsep dasar yang dirujuk sebagai acuan dasar dalam melakukan penelitian ini.

### **BAB III ANALISIS DAN PEMETAAN DESIGN PATTERN DAN CLEAN CODE**

BAB III membahas mengenai tahapan analisis dari objek penelitian untuk mengetahui masalah apa yang saja yang ditemukan dari penelitian yang dilakukan dan menemukan solusi yang cocok terhadap temuan masalah.

#### **BAB IV REFACTORING DAN PENGUJIAN MAINTAINABILITY**

BAB IV berisi mengenai implementasi *refactoring* dari hasil analisis pada BAB III terhadap objek penelitian. Penentuan strategi implementasi dan pengujian juga akan dibahas pada bab ini.