

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

PT Svara Inovasi Indonesia merupakan perusahaan *startup* yang bergerak di bidang media, *broadcasting*, dan *community platform*. Salah satu sistem informasi yang telah dikembangkan oleh PT Svara adalah sistem informasi radio, musik, berita, dan sosial media dalam bentuk aplikasi bernama Svara berbasis android. Berdasarkan wawancara bersama *project manager* yang terdapat pada [LAMPIRAN A], perusahaan melakukan *development* berdasarkan permintaan dari atasan, *client whitelabeling*, dan user aplikasi svara yang terus memberikan masukan, fitur tambahan, dan penemuan bugs. Sampai saat ini, *developing* dan *maintaining* masih terus dilakukan di aplikasi Svara guna untuk meningkatkan fitur dan mengurangi *bugs* agar aplikasi dapat digunakan dengan baik oleh *client* dan *user* sesuai [LAMPIRAN B].

Menurut sudut pandang *developer* aplikasi dari hasil wawancara pada [LAMPIRAN A], seringkali *developer* kesulitan dalam melakukan *developing* dan *maintaining* dikarenakan tidak adanya dokumentasi kode tertulis untuk dijadikan acuan mempelajari struktur kode yang telah dibuat oleh orang-orang terdahulu. Dalam 6 tahun terakhir semenjak aplikasi dikembangkan, diketahui telah tercatat sebanyak 14 orang terlibat dalam pengembangan aplikasi Svara yang dilampirkan di [LAMPIRAN C]. Dari data tersebut dapat dikatakan bahwa *resource developer* aplikasi sering berganti. Ini mengakibatkan banyaknya *coding style* berbeda di dalam satu aplikasi. *Coding style* yang berbeda ini terjadi dikarenakan tidak adanya *design pattern* utuh yang diimplementasi pada *source code* yang dilampirkan pada [LAMPIRAN D], sehingga membuat *developer* yang bekerja saat ini kesulitan untuk mempelajari struktur kode terdahulu.

Hal lainnya untuk mengetahui *developer* kesulitan dalam *developing* dan *maintaining* adalah dengan melakukan penelitian terhadap kualitas kode. Dalam aplikasi Svara terdapat 1.147 kelas dengan cyclomatic complexity nya sebesar

16,47 yang dilampirkan di [LAMPIRAN D], *maintainability index* sebesar 36,57 yang dilampirkan di [LAMPIRAN D], dan klasifikasi penilaian *reusability* yaitu *medium* yang dilampirkan di [LAMPIRAN D]. Dari data tersebut diketahui bahwa aplikasi Svara memiliki tingkat kompleksitas yang tinggi, *maintainability* yang rendah, dan *reusability* yang belum cukup optimal dan perlu ditingkatkan. Berdasarkan keadaan tersebut, dilakukan penelitian untuk meningkatkan kualitas kode dari aplikasi Svara dengan menggunakan *Software Quality Models* dari McCall yang terdiri dari *Correctness, Efficiency, Flexibility, Integrity, Interprobability, Maintainability, Portability, Reliability, Reusability, Testability,* dan *Understandability* yang mana setiap tujuan tersebut memiliki aspek penilaiannya masing-masing [1]. Sesuai dengan permasalahan yang ada saat ini pada *source code* aplikasi Svara maka penelitian ini akan berfokus pada penilaian *code quality* dengan atribut yang dinilai yaitu *Maintainability* dan *Reusability*.

Penelitian ini akan menghasilkan aplikasi dengan struktur kode yang berbeda dari sebelumnya, karena pada penelitian ini akan dilakukan *reengineering* yaitu proses memperbaharui perangkat lunak tanpa mempengaruhi fungsinya untuk menciptakan struktur kode yang terstruktur dan meningkatkan *code quality* dengan pendekatan prinsip *clean architecture* [2]. Hasil *reengineering* diharapkan dapat meningkatkan *code quality* untuk memudahkan *developer* dalam melakukan *developing* dan *maintaining* aplikasi Svara.

1.2 Identifikasi Masalah.

Berdasarkan latar belakang masalah diatas, maka dapat diidentifikasi permasalahan sebagai berikut :

1. Struktur kode aplikasi Svara menyulitkan pengembang dalam memahami aplikasi.
2. Aplikasi Svara memiliki kode yang menyulitkan pengembang dalam proses pemeliharaan, perbaikan, dan pengembangan lebih lanjut.
3. Aplikasi Svara memiliki arsitektur kode yang kurang mendukung dalam penggunaan kembali kode dan komponen.

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah melakukan *reengineering* dengan pendekatan *clean architecture* pada aplikasi Svara. Sedangkan tujuan dari penelitian ini adalah sebagai berikut :

1. Memudahkan pengembang dalam memahami struktur kode aplikasi Svara.
2. Memudahkan pengembang dalam proses pemeliharaan, perbaikan, dan pengembangan lebih lanjut.
3. Menciptakan arsitektur kode yang mendukung dalam penggunaan kembali kode dan komponen.

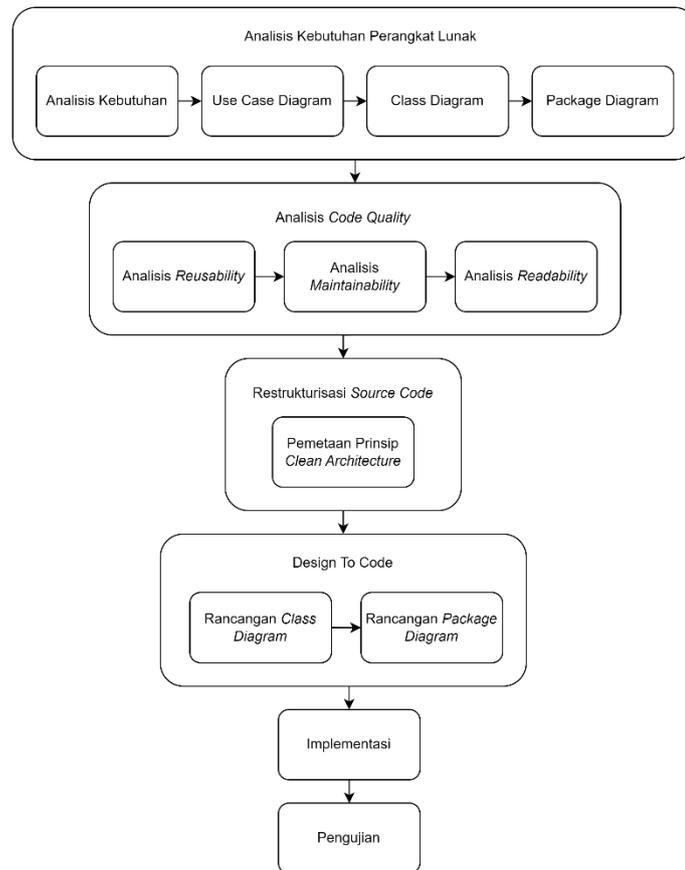
1.4 Batasan Masalah

Agar pembahasan lebih optimal maka penulis membatasi ruang lingkup yang akan dibahas, berikut Batasan masalah dalam penelitian ini adalah:

1. *Reengineering* hanya berfokus pada dua atribut kualitas kode perangkat lunak yaitu Maintainability dan Reusability.
2. Pendekatan analisis perangkat lunak menggunakan *Object Oriented Analysis and Design* (OOAD).

1.5 Metodologi Penelitian

Metodologi penelitian merupakan cara ilmiah dalam upaya menemukan data demi tujuan dan kegunaan tertentu [3]. Jenis penelitian yang dilakukan pada penelitian ini adalah *Enhanced Reengineering* yang mengacu pada jurnal "*Software Re-engineering: An Overview*" [4]. Setelah itu Metodologi tersebut disesuaikan dengan kebutuhan penelitian. Adapun langkah-langkah yang dilakukan pada penelitian ini dapat dilihat pada Gambar 1.1.



Gambar 1.1 Metodologi Penelitian

1. Analisis Kebutuhan Perangkat Lunak

Pada tahap ini, dilakukan analisa terhadap aplikasi Svara untuk memahami spesifikasi kebutuhan perangkat lunak aplikasi. Selain itu, perlu diketahui spesifikasi perangkat lunak apa saja yang disajikan dari hasil menganalisa aplikasi Svara. Analisis kebutuhan ini menggunakan pendekatan OOAD yang meliputi Analisis Data, *Use Case Diagram*, *Class Diagram*, dan *Package Diagram* [5].

2. Analisis Code Quality

Pada tahap ini, dilakukan analisa terhadap *code quality* di aplikasi svara. Analisis disini akan berfokus pada atribut *Maintainability* dan *Reusability*. Maka dari itu, pada tahapan analisis *code quality* ini meliputi Analisis *Maintainability*, Analisis *Reusability* [1].

3. Restrukturisasi *Source Code*

Pada tahap ini, dilakukan pemilihan keputusan mengenai teknologi apa saja yang akan diimplementasi pada *source code* aplikasi Svava dan tidak akan merubah proses bisnis atau fungsionalitas yang ada dengan pendekatan prinsip *clean architecture*. Lalu, Setiap kode program yang terlibat diperbaiki keterbacaannya menggunakan konsep *clean code* dan *SOLID principle* [6].

4. Design To Code

Tahap ini memberikan rincian tentang proses desain ke kode. Pada tahap ini, sesuai dengan dokumen re-desain, kode telah dibuat oleh programmer. Biasanya, algoritma lama diimplementasikan dalam bahasa pengembangan tradisional dengan fungsionalitas yang ada yang perlu ditulis ulang dengan fungsionalitas baru. Pada tahap ini akan dilakukan Perancangan *class diagram* dan *package diagram*.

5. Implementasi

Tahap ini adalah tahap terakhir dari mekanisme *reengineering* aplikasi svava. Pada tahap ini perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program.

6. Pengujian

Pada tahap ini akan dilakukan pengujian kualitas kode aplikasi Svava dengan membandingkan kualitas kode aplikasi Svava yang lama dengan aplikasi Svava yang telah di *reengineering*.

1.6 Sistematika Penulisan

Sistematika penulisan ini disusun untuk memberikan gambaran umum mengenai penulisan laporan skripsi yang akan dilakukan. Sistematika Penulisan laporan skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini berisi uraian latar belakang masalah, rumusan masalah, maksud dan tujuan, batasan masalah, Metodologi penelitian tahap pengumpulan data, model pengembangan perangkat lunak dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini akan membahas berbagai konsep dasar dan teori-teori pendukung yang berhubungan dengan Pembangunan sistem.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan membahas mengenai deskripsi sistem, analisis kebutuhan dalam pembangunan sistem serta perancangan sistem.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini berisi hasil implementasi analisis dari BAB 3 dan perancangan aplikasi yang dilakukan, serta hasil pengujian aplikasi untuk mengetahui apakah aplikasi yang telah di *refactoring* sudah memenuhi kebutuhan.

BAB 5 KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang diperoleh dari hasil pengujian sistem, serta saran untuk pengembangan aplikasi untuk masa yang akan datang.