BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

PD. Fahmi Jaya merupakan toko yang menjual berbagai macam kebutuhan pokok dalam harga grosir atau eceran. PD. Fahmi Jaya memiliki aplikasi *Point of Sales* (POS) yang berbasis android. Aplikasi ini menangani pengelolaan data penjualan dan pembelian produk, pendataan riwayat penjualan dan pembelian, pengelolaan data pelanggan, pengelolaan stok produk, melihat profil, dan pengaturan aplikasi untuk koneksi ke *printer*.

Aplikasi POS yang berbasis android pada PD. Fahmi Jaya masih dilakukan pengembangan hingga saat ini. Berdasarkan wawancara awal, pengembangan dilakukan sebanyak satu sampai dua kali dalam satu tahun dan kelas yang ada menghambat proses pengembangan. Setelah melakukan inspeksi kelas, ternyata ditemukan masalah bahwa kelas yang ada pada saat ini sulit dipahami dan digabungnya kelas yang menangani tampilan dan proses bisnis. Kelas yang sulit dipahami akan berdampak buruk karena beberapa pengembang aplikasi tidak bisa memahaminya lagi setelah enam bulan. Maka dari itu kelas yang dapat dipahami oleh pengembang aplikasi lainnya adalah hal penting [1]. Kelas yang menangani tampilan dan proses bisnis merupakan kelas yang memiliki tanggung jawab atau unit besar. Sementara itu indikasi dari kode yang dapat dipelihara adalah kelas yang memiliki tanggung jawab yang kecil [2]. Apabila terus menerus dibiarkan, maka pengembangan selanjutnya dapat terhambat [1] [2]. Adapun hasil inspeksi kelas dapat dilihat pada **LAMPIRAN A** HASIL INSPEKSI SOURCE CODE.

Berdasarkan fenomena tersebut, dilakukan riset untuk meningkatkan kualitas *source code* dari aplikasi POS. Menurut jurnal yang berjudul "*Software Re-engineering: An Overview*" bahwa terdapat tiga metode untuk melakukan *re-engineering* yaitu *Reverse Engineering, Forward Engineering* , dan *Enhanced Re-engineering*. Berdasarkan jurnal tersebut, disebutkan bahwa *Enhanced Re-engineering* dapat memberikan peningkatan pada kualitas *source code* yang telah

dilakukan *re-engineering* [3]. Menurut buku *Software Engineering* karangan dari Roger S. Pressman pada edisi ketujuh bahwa *Software Quality Assurance* (SQA) dibagi menjadi beberapa tujuan yaitu *Requirements Quality*, *Design Quality*, Code *Quality*, dan *Quality Control Effectiveness* yang mana setiap tujuan tersebut memiliki aspek penilaiannya masing-masing. Pada tujuan *Code Quality*, terdapat lima atribut yang dinilai yaitu *Complexity*, *Mainainability*, *Understandability*, *Reusability*, dan *Documentation* [4]. Juga, menurut jurnal yang berjudul "*Framework* dan *Code Generator* Pengembangan Aplikasi Android dengan Menerapkan Prinsip *Clean Architecture*" dengan menerapkan prinsip-prinsip dari *Clean Architecture* dapat meningkatkan kualitas dari kelas aplikasi[5].

Berdasarkan masalah yang ada saat ini pada kelas aplikasi POS berbasis android maka *re-engineering* atau rekayasa ulang sistem merupakan solusi karena dapat memberikan peluang dalam meningkatnya *code quality* dan membuat aplikasi lebih mudah untuk dilakukan pemeliharaan [3]. *Re-engineering* yang akan dilakukan adalah pembaharuan struktur *source code* yang sudah dibangun dengan menerapkan prinsip-prinsip dari *Clean Architecture*. Sejalan dengan permasalahan yang ada penelitian ini akan berfokus pada salah satu tujuan dari SQA yaitu *Code Quality* dan Atribut yang dinilai pada *Code Quality* adalah *Maintainability* dan *Documentation* [4]. Hasil dari *Re-engineering* diharapkan dapat meningkatkan *code quality* dan membuat aplikasi *Point of Sales* berbasis android ini dapat dilakukan pemeliharaan dengan baik.

1.2 Perumusan Masalah

Berdasarkan penjelasan dari latar belakang masalah maka perumusan masalah pada penelitian ini adalah apakah dengan menerapkan konsep dari *clean architecture* dapat meningkatkan *code quality*.

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah untuk melakukan *re-engineering source code* aplikasi POS berbasis android PD. Fahmi Jaya dengan menerapkan konsep dari *Clean Archite*cture. Tujuan dari penelitian ini yaitu meningkatkan *code quality* pada *source code* POS.

1.4 Batasan Masalah

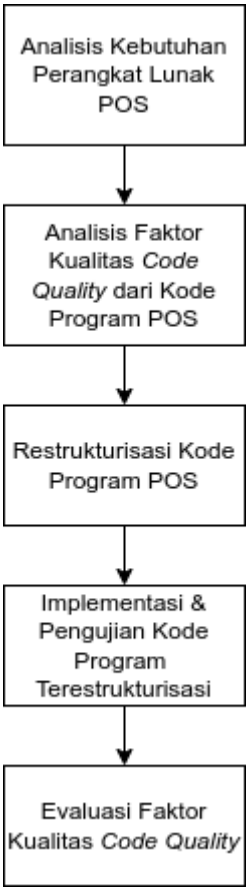
Dalam penelitian ini, terdapat beberapa batasan masalah agar penelitian ini tidak menyimpang dari tujuan yang ingin dicapai. Berikut batasan masalah pada penelitian ini:

1. *Re-engineering* yang dilakukan hanya berfokus pada dua dari lima atribut dalam *code quality*, yakni: *Maintainability*, dan *Documentation* [4].
2. Pendekatan analisis aplikasi menggunakan *Object Oriented Analysis and Design* (OOAD).

1.5 Metodologi Penelitian

Jenis penelitian pada penelitian ini adalah penelitian kualitatif *case study* [8]. Metodologi yang digunakan adalah *Enhanced Reenginering* yang mengacu pada jurnal "*Software Re-engineering: An Overview*" [3]. Metodologi tersebut selanjutnya disesuaikan dengan kebutuhan penelitian. Penyesuaian metodologi meliputi tidak dilakukannya tahapan yang bernama "Membangun Kembali Jika Teknologi yang Diusulkan Lebih Baik dari Sistem yang Ada". Tahap ini tidak cocok dengan kebutuhan penelitian karena tidak sejalan dengan tujuan penelitian. Adapun langkah-langkah yang dilakukan pada penelitian ini dapat dilihat pada **Gambar 1-1** Metodologi Penelitian.

.



Gambar 1-1 Metodologi Penelitian

1. **Analisis Kebutuhan Perangkat Lunak**

   Pada tahap ini, analisis kebutuhan perangkat lunak dilakukan untuk memeriksa kebutuhan pada aplikasi POS berbasis android. Analisis kebutuhan ini menggunakan pendekatan OOAD meliputi *Use Case Diagram*, Analisis Data, *Package Diagram*, dan *Class Diagram* [9].

2. **Analisis Faktor Kualitas *Code Quality* dari Kode Program POS**

   Tahapan ini dilakukan untuk mendapatkan gambaran mengenai kondisi kelas aplikasi POS saat ini. Analisis ini akan berfokus pada dua dari lima atribut dalam *code quality*, yakni: *Maintainability* dan *Documentation* [4].

3. **Restrukturisasi Kode Program**

   Tahap ini adalah tahap pada saat kode program direstrukturisasi. Restrukturisasi ini dilakukan dengan menerapkan prinsip-prinsip dari *clean architecture* [10]. Google juga menyarankan penggunaan *ViewModel*

dalam *source code* aplikasi android karena dapat mempertahankan *UI-related data* ketika terjadi perubahan konfigurasi seperti orientasi layar [6]. Aplikasi POS berbasis android ini pun akan dilakukan migrasi pustaka ButterKnife ke *view binding*. Mengingat pengembangan pustaka ButterKnife telah dihentikan pengembangannya dan Jake Wharton selaku pengembang dari ButterKnife menyarankan untuk melakukan migrasi ke *view binding* [7]. Tahap restrukturisasi digambarkan dengan *package diagram* dan *class diagram* [9].

4. **Implementasi & Pengujian Kode Program Terestrukturisasi**

   Tahap ini adalah tahap terakhir dari mekanisme *re-engineering* pada aplikasi POS berbasis android. Pada tahap implementasi, bagian-bagian tertentu diganti berdasarkan pada tahap restrukturisasi kode program. Setelah diimplementasi, maka dilakukan pengujian untuk memastikan semua fungsi berjalan sama dengan aplikasi yang belum dilakukan *re-engineering*.

5. **Evaluasi Faktor Kualitas *Code Quality***

   Tahap ini merupakan lanjutan mengenai evaluasi kode program yang telah di restrukturisasi. Dalam tahapan ini dilakukan pengukuran ulang sebanyak dua dari lima atribut dalam *code quality*, yakni: *Maintainability* dan *Documentation* [4].

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 PERANCANGAN

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 IMPLEMENTASI DAN PENGUJIAN

BAB IV berisi tentang implementasi yang telah dilakukan berdasarkan hasil analisis sebelumnya. Pada bagian ini juga akan dibahas bagaimana hasil pengujian dari implementasi yang baru dan perbandingan dari hasil pengujian yang lama.

BAB V KESIMPULAN DAN SARAN

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