

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

PDAM Sumedang adalah perusahaan yang bergerak dalam bidang pengelolaan air bersih dan distribusi air minum yang berada di Sumedang, Jawa Barat, Indonesia. Pada tahun 2018 PDAM Sumedang membuat sistem pencatatan yaitu AuroraViews. Sistem ini dapat meningkatkan efisiensi dan akurasi dalam pencatatan data, serta memudahkan dalam melakukan analisis data. Selain itu, sistem baru ini juga dapat meningkatkan keamanan data dan mengurangi risiko *human error*.

Berdasarkan wawancara dengan pengembang, sistem AuroraViews masih dalam proses pengembangan hingga saat ini dengan frekuensi pengembangan sekitar satu atau dua kali dalam setahun. Dalam proses pengembangannya, ada beberapa masalah yang dihadapi, seperti kesulitan dalam memahami kelas yang menghambat proses pengembangan, kode yang diberi komentar tetapi tidak menjelaskan secara detail, serta instalasi dan pengaturan framework yang rumit dan menghambat proses pergantian pengembang dari yang lama ke yang baru. Setelah permasalahan diketahui maka dilakukan analisis pada sistem AuroraViews dengan menggunakan PHPMetrics dari analisis ditemukan rendahnya *Maintainability Indeks* dan beberapa kelas yang memiliki *Cyclomatic Complexity* yang tinggi. Kelas yang sulit dipahami dapat menimbulkan masalah karena pengembang sistem mungkin tidak lagi memahaminya setelah enam bulan. Oleh karena itu, memiliki kelas yang dapat dipahami oleh pengembang lain terutama pengembang baru menjadi sangat penting [1]. Hal-hal tersebut apabila dibiarkan dapat mengakibatkan proses pemeliharaan dan pengembangan selanjutnya menjadi terhambat [2]. Adapun bukti dari hasil analisis dapat dilihat pada Error! Reference source not found..

Setelah mengamati masalah yang terjadi, studi literatur telah dilakukan untuk mengevaluasi kualitas perangkat lunak. Temuan dari studi tersebut mengindikasikan bahwa kualitas perangkat lunak dapat dilihat dari tiga perspektif, seperti yang telah diusulkan oleh McCall. Poin penting yang dicetuskan oleh McCall adalah untuk menilai kualitas perangkat lunak dan meningkatkannya [3]. Penelitian ini menilai perangkat lunak dari perspektif *Product Revision* dengan memfokuskan pada faktor *maintainability*. Metode yang digunakan adalah McCall dengan mempertimbangkan faktor Product Revision untuk meningkatkan *indeks maintainability* dan meningkatkan nilai *readability* serta memastikan bahwa perangkat lunak dapat mudah dipelihara dan dimodifikasi dalam jangka waktu yang panjang [4]. Faktor *maintainability* digunakan untuk mengevaluasi seberapa mudah suatu sistem dapat dipelihara, termasuk penulisan dokumentasi dan kode sumber yang baik. Dalam menulis kode, konsep *Clean Code* dan *Design Pattern* sangat penting untuk diterapkan agar keterbacaan dan struktur kode sumber menjadi lebih baik, sehingga memudahkan pemeliharaan. [5].

Untuk meningkatkan *maintainability* dan *readability* pada sistem AuroraViews, konsep *Clean Code* dan *Design Pattern* akan diterapkan. Tujuan dari *Clean Code* adalah untuk meningkatkan keterbacaan kode sehingga lebih mudah dipahami dan dipelihara, sedangkan tujuan dari *Design Pattern* adalah untuk memperbaiki struktur modul dalam sistem.

1.2 Perumusan Masalah

Berdasarkan latar belakang, maka dapat diidentifikasi masalah pada penelitian ini adalah apakah dengan menerapkan *Clean Code* dan *Design Pattern* dapat meningkatkan *maintainability* dan *readability* pada sistem AuroraViews?

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah melakukan implementasi *Clean Code* dan *Design Pattern* pada sistem AuroraViews. Sedangkan tujuan dari penelitian ini adalah untuk membantu pengembang dalam meningkatkan *maintainability* dan *readability* pada sistem AuroraViews.

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini agar menyimpang dari tujuan, sebagai berikut :

1. Penelitian yang dilakukan hanya meliputi kode dan dokumentasi dari sistem AuroraViews.
2. Analisis dan paradigma pemrograman menggunakan pendekatan berorientasi objek.
3. Penulisan kode program menggunakan framework Laravel.
4. Proses refactoring hanya pada *class* yang berjenis seperti *controller* atau *class* yang mengelola logika bisnis, akses data, dan permintaan, sementara model tidak terlibat.

1.5 Metodologi Penelitian

Penelitian ini menggunakan metodologi kualitatif yang mengacu pada buku *Refactoring to Pattern*, dan metodologi penelitian akan disesuaikan dengan kebutuhan penelitian. Berikut adalah diagram secara umum yang menggambarkan tahapan penelitian yang akan dilakukan dapat dilihat pada **Gambar 1.1**.



Gambar 0.1 Tahapan Penelitian

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

a. Analisis Kebutuhan Perangkat Lunak AuroraViews

Pada tahap ini, dilakukan analisis kebutuhan perangkat lunak untuk memeriksa kebutuhan sistem AuroraViews. Pendekatan OOAD digunakan dalam analisis kebutuhan ini, termasuk *Use Case Diagram*, Analisis Data, *Package Diagram*, dan *Class Diagram* [6].

b. Analisis Faktor Kualitas *Code Quality* dari Kode Program AuroraViews

Tahap ini dilakukan untuk mengumpulkan informasi tentang kondisi kelas dalam sistem AuroraViews melalui analisis Product Revision yang berfokus pada *maintainability* dan *readability*.

c. Pemetaan Konsep *Clean Code & Design Pattern*

Tahap ini melibatkan pemetaan pada kode program untuk meningkatkan kualitas dan memudahkan pemeliharaan serta pengembangan kode program. Pemetaan konsep *clean code* bertujuan untuk menciptakan kode program yang mudah dipahami, mudah dirawat, dan memiliki struktur yang baik. Sedangkan pemetaan konsep *design pattern* bertujuan untuk membangun arsitektur yang baik dan menggunakan pola desain yang telah terbukti efektif dalam menyusun kode program.

d. Implement & Pengujian Kode Program Terstrukturisasi

Tahap implementasi melibatkan penggantian bagian-bagian tertentu dalam kode program yang telah direstrukturisasi. Setelah proses penggantian selesai, dilakukan pengujian untuk memastikan bahwa semua fungsi masih berjalan dengan baik seperti sebelumnya.

e. Evaluasi Faktor Kualitas Code Quality

Tahap terakhir adalah evaluasi terhadap kode program yang telah direstrukturisasi. Pada tahap ini, dilakukan pengukuran ulang terhadap tingkat *maintainability* dan *readability* untuk memastikan bahwa perbaikan yang dilakukan berhasil meningkatkan kemudahan pemeliharaan kode program.

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.