BAB 1
PENDAHULUAN

1.1 Latar Belakang

Pengembangan perangkat lunak pada umumnya sulit untuk diselesaikan dengan tepat waktu. Masalah ini umumnya terjadi karena adanya perubahan spesifikasi dari sebuah perangkat lunak dan *programmer* mengalami kesulitan dalam memenuhi perubahan spesifikasi yang terjadi. Kesulitan ini umumnya terjadi karena kode program dari sebuah perangkat lunak memiliki nilai metrik *software metrics* yang rendah, sehingga dalam melakukan perubahan perangkat lunak, kode program dibuat kembali dari nol.

Penggunaan *software architecture* adalah salah satu cara untuk meningkatkan nilai *software metrics* dari sebuah perangkat lunak, sehingga perubahan spesifikasi dapat dengan mudah dilakukan dan kode program yang telah dibuat dapat digunakan kembali pada pengembangan perangkat lunak yang lain. *Software architecture* adalah sebuah fondasi dari perangkat lunak yang mengkomunikasikan antara elemen perangkat lunak secara terstruktur. Terdapat beberapa *software architecture* pada perangkat lunak, beberapa contohnya pada *iOS* adalah Model-View-Controller dan Clean Swift.

Penelitian ini akan menguji nilai dari *software metrics* dari sebuah perangkat lunak dengan menggunakan arsitektur yang berbeda. Nilai *software metrics* yang didapatkan akan dianalisis untuk menentukan arsitektur yang sebaiknya digunakan dalam pengembangan perangkat lunak. Penelitian ini diharapkan dapat membantu *programmer* dalam menggunakan *software architecture* yang tepat sehingga pengembangan perangkat lunak dapat diselesaikan tepat waktu.

1.2 Rumusan Masalah

Berdasarkan masalah yang didapat dari kesulitan *programmer* dalam mengembangkan perangkat lunak, maka dapat dirumuskan rumusan masalah menjadi sebagai berikut:

1. Bagaimana cara menentukan *software architecture* yang tepat dan sesuai dengan kebutuhan perangkat lunak?

1.3 Tujuan Pembahasan

Berdasarkan rumusan masalah yang ada pada subbab 1.2, terdapat beberapa tujuan pembahasan sebagai berikut:

1. Menganalisis beberapa *software arhictecture* berdasarkan nilai *software metrics* dengan spesifikasi kebutuhan perangkat lunak tertentu.
2. Memberikan saran pemilihan arsitektur yang sesuai dengan kebutuhan.

1.4 Ruang Lingkup

Ruang lingkup tugas akhir ini terbatas pada:

1. Aplikasi *iOS* dengan menggunakan studi kasus perusahaan X pada proses yang berkait dengan data perusahaan X.
2. Menggunakan *tools lizard* sebagai pengukur nilai *software metrics*.
3. Pengujian *software architecture* MVC dan Clean Swift.

1.5 Sumber Data

Sumber data tugas akhir ini bersumber pada:

1. Jurnal, buku, dan halaman web yang berhubungan dengan rekayasa perangkat lunak mengenai arsitektur perangkat lunak, *software metrics*, dan arsitektur *mobile*.
2. Studi kasus perusahaan X yang berhubungan dengan spesifikasi aplikasi yang dibutuhkan dan data perusahaan X untuk pembuatan aplikasi sebagai bahan studi pengujian *software metrics*.

1.6 Sistematika Penyajian

Sistematika pembahasan yang akan digunakan dalam laporan ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan pembahasan, ruang lingkup, sumber data, dan sistematika penyajian dari tugas akhir ini.

BAB II. KAJIAN TEORI

Bab ini akan menjelaskan teori yang berhubungan dengan proses analisis, desain, *web services,* arsitektur perangkat lunak, dan nilai *software metrics* dari aplikasi.

BAB III. ANALISIS DAN RANCANGAN

Bab ini akan berisi profil perusahaan*, Unified Modelling Language Diagram,* rancangan desain tampilan aplikasi, dan metode pengujian *software metrics.*

BAB IV. IMPLEMENTASI

Bab ini akan berisi implementasi arsitektur Model-View-Controller dan Clean Swift, metode pengujian *complexity* dan *Simple Component Cohesion Metrics.*

BAB V. PENGUJIAN

Bab ini akan berisi hasil *benchmarking* pengujian nilai *complexity* dan *Simple Component Cohesion Metrics*.

BAB VI. SIMPULAN DAN SARAN

Bab ini akan berisi simpulan dari hasil penelitian yang dilakukan dan saran untuk penelitian dengan topik serupa.