

BAB I

PENDAHULUAN

1.1. Latar Belakang

Clean architecture adalah sebuah konsep atau metode pengembangan aplikasi yang berfokus pada pemisahan bagian kode tertentu yang terbagi dalam beberapa lapis kode berdasarkan tujuan dari kode tersebut, sehingga satu bagian dan bagian lain sedikit atau bahkan tidak memiliki ketergantungan (Robert C. Martin, 2012). Konsep *clean architecture* dibuat oleh Robert C. Martin yang ditulis pada blog pribadinya. *Clean architecture* memiliki satu aturan pasti, yaitu sebuah kode pada lapisan dalam tidak boleh bergantung kepada lapisan luarnya. Dalam implementasinya, *clean architecture* dapat dilakukan dengan menerapkan sebuah *design pattern* sebagai metode untuk menerapkan *clean architecture* dengan baik.

Design pattern atau yang dapat diartikan sebagai pola desain adalah metode yang dibuat untuk membantu tim pengembang dalam menemukan solusi dari masalah-masalah umum yang muncul saat pengembangan perangkat lunak sedang berlangsung. *Design pattern* bukanlah suatu metode yang dapat diimplementasikan langsung menjadi sebuah kode program, tetapi merupakan sebuah pola untuk menyelesaikan masalah dalam situasi yang bermacam-macam (Roni Setiawan, 2021).

Pengembangan aplikasi android sudah dilakukan cukup lama, dalam pengembangannya tercipta berbagai macam model arsitektur yang dapat diterapkan. Sistem operasi android sudah semakin berkembang sampai pada titik di mana diperlukan sebuah arsitektur baru yang dapat mendukung kinerja sistem operasi android dengan baik. Beberapa arsitektur populer yang sering digunakan dalam pengembangan aplikasi Android yang menjadi fokus utama dalam penelitian ini yaitu MVP (*Model View Presenter*), MVI (*Model View Intent*), dan MVVM (*Model View ViewModel*).

Penelitian terkait telah dilakukan dengan membandingkan jenis arsitektur yang cukup berbeda, yaitu MVC, MVP dan MVVM yang ditulis pada tesis berjudul “A comparison of Android native app architecture MVC, MVP and MVVM” oleh Lou T. pada tahun 2016. Berbagai jenis arsitektur yang telah dikembangkan tentu memiliki kekurangan dan kelebihan masing-masing yang terkadang membuat bingung seorang pengembang dalam menentukan arsitektur yang sesuai dengan jenis

aplikasinya. Pada penelitian ini dilakukan perbandingan ketiga jenis arsitektur tersebut. Perbandingan dilakukan dengan beberapa jenis parameter, seperti modifiabilitas, testabilitas, dan performa berdasarkan skenario tertentu untuk menentukan arsitektur mana yang paling sesuai dengan jenis pengembangan aplikasi tertentu.

1.2. Rumusan Masalah

Rumusan masalah yang didapat dari latar belakang tersebut adalah sebagai berikut:

1. Apa perbedaan antara arsitektur MVP, MVI, dan MVVM?
2. Arsitektur mana yang terbaik untuk masing masing aspek mulai dari modifiabilitas, testabilitas, dan performa?
3. Bagaimana cara menentukan jenis arsitektur terbaik untuk pengembangan jenis aplikasi tertentu?

1.3. Batasan Masalah

Terdapat batasan masalah pada penelitian ini, yaitu:

1. Perbandingan hanya dilakukan dengan 3 jenis arsitektur populer pada pengembangan aplikasi Android yaitu MVP, MVI, dan MVVM.
2. Implementasi sistem pada pengembangan aplikasi Android dilakukan dengan menggunakan Android Studio dan Kotlin sebagai bahasa pemrogramannya.
3. Perancangan sistem dilakukan dengan membuat aplikasi sederhana untuk menampilkan sebuah data dari internet.
4. Pengujian sistem dilakukan dengan *unit test* untuk menguji masing masing fungsi pada kode program.
5. Perbandingan yang diuji dan dianalisis adalah tingkat modifiabilitas, testabilitas, dan performa.

1.4. Tujuan

Adapun tujuan dari penelitian yang dilakukan ini adalah:

1. Mengetahui perbedaan antara arsitektur MVP, MVI, dan MVVM dengan tepat.
2. Mengetahui jenis arsitektur terbaik untuk aspek modifiabilitas, testabilitas, dan performa.
3. Membantu pengembang aplikasi Android menentukan jenis arsitektur yang sesuai dengan kebutuhan aplikasi yang akan dikembangkan berdasarkan hasil kesimpulan penelitian ini.