

BAB I PENDAHULUAN

1.1 Latar Belakang

Pada Era revolusi industri 4.0 pengembangan produk digital seperti aplikasi *mobile* harus bisa dengan cepat beradaptasi dan merespon segala perubahan yang terjadi, hal itu harus dilakukan agar produk aplikasi bisa tetap bersaing dan memenuhi kebutuhan pasar [1]. Namun untuk menciptakan aplikasi demikian bukanlah hal yang mudah, karena ada beberapa aktivitas yang harus dilakukan seperti penambahan fitur, memaksimalkan kode program, perbaikan error, dan lain sebagainya. Oleh karena itu, sangatlah penting untuk melakukan optimasi atau peningkatan kualitas aplikasi dengan meningkatkan *maintainability* dan mengoptimalkan *code complexity* [2]. Dengan begitu kode program akan lebih *maintainable* dan fleksibel, sehingga akan lebih mudah melakukan perubahan pada kode [2]. Manajemen file yang baik dan penulisan *kode clean* sangatlah penting diterapkan dalam pengembangan aplikasi terutama aplikasi *mobile*, agar dapat memudahkan maintenance dan meningkatkan fleksibilitas aplikasi yang dikembangkan [3].

Terdapat banyak pola *architecture* yang biasa dipakai untuk mengembangkan aplikasi *mobile* [4], [5]. Salah satunya adalah *clean architecture*. *Clean Architecture* adalah salah satu pola arsitektur aplikasi yang bertujuan untuk memisahkan setiap lapisan penting dalam aplikasi sesuai kegunaannya. Pemisahan lapisan tersebut bertujuan agar tidak terjadi ketergantungan antara setiap lapisan aplikasi [6]. Selain itu dengan adanya sejumlah aturan pada *clean architecture* membuat struktur project aplikasi akan lebih terstruktur dan kode yang ditulis pun akan menjadi lebih sederhana sehingga mudah dibaca. Dengan begitu dengan menerapkan *clean architecture* aplikasi yang dihasilkan akan mudah di-*maintain* dan lebih fleksibel.

Dalam menganalisis penerapan *Clean architecture* terhadap *maintainability* dan *complexity code*, peneliti akan mengimplementasikan *Clean architecture* pada

studi kasus yang akan dijadikan baseline penelitian ini. Adapun studi kasusnya adalah pengembangan aplikasi *Intention Habit* yang merupakan project aplikasi milik peneliti. *Intention habit* merupakan aplikasi yang membantu pengguna dalam mencapai tujuan yang ingin dicapai dengan cara *goal setting* dan membangun *habit* yang sejalan dengan tujuan yang dicapai.

Dalam pengembangan aplikasi *mobile* tersebut, peneliti akan menggunakan *flutter* sebagai *framework* utama pengembangan aplikasi *mobile* ini. *Flutter* merupakan *framework* aplikasi *multi-platform open-source* yang memungkinkan digunakan untuk mengembangkan aplikasi *Android*, *iOS*, *Web*, Bahkan *desktop* hanya dengan satu basis kode saja [7]. Selain itu *flutter* juga memiliki fitur unggulan yaitu *Hot reload* yang memungkinkan pengembang dapat dengan cepat melihat hasil perubahan kode. Dengan karakteristiknya yang demikian *flutter* sangatlah cocok digunakan untuk mengembangkan aplikasi *mobile* dan mengadopsi penggunaan *clean architecture* untuk menghasilkan aplikasi yang *maintainable* dan *fleksibel*. Dan untuk mengukur pengaruh penerapan *clean architecture* terhadap *maintainability* dan *complexity code* bisa dilakukan dengan menggunakan bantuan *package Dart code metrics*. Yang akan bekerja mengumpulkan *data* analisis pada kode program.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan *clean architecture* dalam pengembangan aplikasi *mobile* berbasis *flutter*?
2. Bagaimana nilai *maintainability* aplikasi yang dikembangkan dengan *clean architecture*?
3. Bagaimana nilai kompleksitas kode aplikasi *mobile* yang dikembangkan dengan *clean architecture* ?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang sudah dipaparkan, maka permasalahan akan dibatasi pada:

1. Penelitian ini hanya akan membahas penerapan *clean architecture* pada pengembangan aplikasi *mobile* berbasis *Flutter*.
2. Studi kasus yang digunakan dalam penelitian ini adalah aplikasi *Intention Habit* milik peneliti.
3. Dalam pengembangan aplikasi *Intention Habit*, penelitian ini akan fokus pada fitur-fitur utama yang meliputi operasi *Create*, *Read*, *Update*, dan *Delete (CRUD) data*, dan tidak akan membahas fitur tambahan atau opsional yang mungkin ada.
4. Analisis yang dilakukan hanya berkaitan dengan *maintainability* dan *complexity code* pada aplikasi yang dikembangkan.
5. Penelitian ini menggunakan *framework Flutter* untuk pengembangan aplikasi *mobile*.
6. Aplikasi *mobile* yang dikembangkan hanya untuk platform *Android*.
7. Penelitian ini menggunakan package *Dart code metrics* untuk mengukur *maintainability* dan *complexity code* pada aplikasi.
8. Penelitian ini terbatas pada penerapan *clean architecture* dalam konteks pengembangan aplikasi *mobile* berbasis *Flutter*, dan tidak mencakup pola arsitektur lain yang mungkin digunakan dalam pengembangan aplikasi tersebut.
9. Penelitian ini menggunakan *MVVM (Model-View-ViewModel)* sebagai *design pattern* dalam konteks *clean architecture* pada pengembangan aplikasi *mobile* berbasis *Flutter*.
10. Penelitian ini akan mencakup penggunaan *dependency injection* sebagai teknik dalam penerapan *clean architecture* pada pengembangan aplikasi *mobile* berbasis *Flutter* untuk meningkatkan modularitas dan mengurangi

ketergantungan antar komponen.

1.4 Tujuan Penelitian

Berdasarkan perumusan masalah yang telah dibuat, tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis dan menjelaskan penerapan *clean architecture* dalam pengembangan aplikasi *mobile* berbasis *Flutter*.
2. Menilai tingkat *maintainability* aplikasi *mobile* yang dikembangkan dengan *clean architecture*.
3. Menilai tingkat kompleksitas kode aplikasi *mobile* yang dikembangkan dengan *clean architecture*.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian yang ingin dicapai, penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Memberikan kontribusi pada pengembangan aplikasi *mobile* berbasis *Flutter* dengan menerapkan *clean architecture* sehingga dapat meningkatkan *maintainability* dan mengurangi kompleksitas kode.
2. Menjadi referensi bagi pengembang aplikasi *mobile* dalam menerapkan *Clean architecture* untuk meningkatkan kualitas aplikasi yang dikembangkan.
3. Memberikan pemahaman yang lebih baik tentang *Clean architecture* dan pengaruhnya terhadap *maintainability* dan kompleksitas kode pada pengembangan aplikasi *mobile* berbasis *Flutter*.
4. Memberikan informasi dan masukan bagi peneliti dan praktisi di bidang pengembangan aplikasi *mobile* dan teknologi informasi dalam memperbaiki kualitas dan efektivitas pengembangan aplikasi *mobile* berbasis *Flutter*.

1.6 Sistematika Penulisan

Dalam laporan penelitian ini terdapat lima bab dengan beberapa pokok bahasan. Oleh karena itu, berikut adalah sistematika penulisan yang ada untuk mempermudah pembaca melihat dan mengetahui skripsi ini secara menyeluruh:

BAB I PENDAHULUAN

Bab ini berisi Latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tinjauan Pustaka yang berisi ulasan dan hasil dari penelitian sebelumnya yang berkaitan dengan penelitian ini, serta dasar-dasar teori yang digunakan dalam penelitian ini.

BAB III METODE PENELITIAN

Bab ini menjelaskan tinjauan umum tentang objek penelitian, alur penelitian, serta alat yang akan digunakan dalam penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

Bab ini merupakan tahapan yang penulis lakukan dalam mengembangkan aplikasi, analisis hingga penerapan aplikasi di objek penelitian.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan.