

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pemilihan pola arsitektur yang tepat sangat penting dalam pengembangan aplikasi iOS untuk memastikan kualitas, *maintainability*, dan efisiensi jangka panjang (Fauzi dkk. 2019). Dengan lebih dari 1,46 miliar perangkat iPhone aktif secara global (Akhtar & Ghafoor, 2021), iOS menjadi platform strategis bagi pengembang aplikasi *mobile*.

Dua pola arsitektur yang umum digunakan adalah Model-View-Controller (MVC) dan Model-View-Presenter (MVP), yang masing-masing menawarkan pendekatan berbeda dalam pemisahan tanggung jawab komponen.

Maintainability aplikasi dapat diukur secara kuantitatif menggunakan metode Maintainability Index dari Code Metrics milik Microsoft, metode ini dipilih karena sudah sangat umum digunakan dalam pengembangan perangkat lunak untuk mengukur tingkat *maintainability* suatu program yang dibangun. Bahkan Microsoft mengadopsinya menjadi *tools* pengukuran *maintainability* dalam VSCode.

Pada tahun 2021, terdapat penelitian yang mengukur *maintainability* dari sebuah sistem yang menerapkan arsitektur MVC dan tanpa menerapkan arsitektur sama sekali menggunakan metode Cyclomatic Complexity, mengukur Halstead Volume serta mengukur metrik Lines of Codes yang semuanya itu merupakan variabel dari metode Maintainability Index (Nahhas, 2021).

Berdasarkan hal tersebut, penelitian ini mencoba membandingkan tingkat *maintainability* aplikasi iOS berbasis Swift dengan *framework* UIKit terhadap pola arsitektur MVC dan MVP menggunakan metode Maintainability Index dari Code Metrics. Selain itu, guna memvalidasi keakuratan Maintainability Index dalam melakukan pengukuran, perbandingan juga akan dilakukan dengan mengukur kecepatan peluncuran, pengujian, serta konsumsi memori dari masing-masing pola arsitektur.

1.2 Rumusan Masalah

Dari latar belakang di atas, maka diperoleh rumusan masalah sebagai berikut ini :

- a. Bagaimana hasil implementasi pola arsitektur MVC dan MVP pada aplikasi iOS berbasis Swift dengan *framework* UIKit?
- b. Pola arsitektur mana yang memiliki *maintainability* yang lebih baik dalam penerapannya pada aplikasi iOS berbasis Swift dengan *framework* UIKit berdasarkan hasil pengukuran dengan metode Maintainability Index dari Code Metrics?
- c. Bagaimana cara melakukan perbandingan tingkat *maintainability* antara pola arsitektur MVC dan MVP secara kuantitatif menggunakan metode Maintainability Index, serta bagaimana tingkat akurasi dari hasil perbandingan tersebut?
- d. Bagaimana rekomendasi implementasi pola arsitektur yang lebih *maintainable* untuk pengembangan aplikasi iOS berbasis Swift dengan *framework* UIKit?

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah di atas, tujuan penelitian ini adalah:

- a. Mengetahui bagaimana hasil implementasi pola arsitektur MVC dan MVP pada aplikasi iOS berbasis Swift dengan *framework* UIKit.
- b. Mengevaluasi dan membandingkan hasil pengukuran Code Metrics untuk kedua pola arsitektur, serta mengidentifikasi kelebihan dan kekurangannya terhadap *maintainability* aplikasi iOS berbasis Swift dengan *framework* UIKit.
- c. Untuk menjelaskan cara melakukan perbandingan secara kuantitatif antara pola arsitektur MVC dan MVP, serta menganalisis tingkat akurasi dari hasil perbandingan tersebut.
- d. Memberikan rekomendasi bagi pengembang dalam memilih pola arsitektur yang lebih baik dari segi *maintainability* berdasarkan hasil penelitian.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberi manfaat sebagai berikut :

- a. Bagi para akademisi, penelitian ini diharapkan dapat memberikan kontribusi sebagai bahan bacaan ilmiah dalam dunia akademik.
- b. Bagi para peneliti, penelitian ini diharapkan dapat menjadi sumber informasi dalam melakukan penelitian lebih lanjut terkait topik serupa.
- c. Bagi para praktisi, penelitian ini diharapkan dapat memberikan referensi pola arsitektur dalam membangun aplikasi iOS berbasis Swift dengan *framework* UIKit.

1.5 Batasan Masalah

Agar pembahasan dalam penelitian ini lebih jelas dan terarah maka pembahasan dibatasi pada:

- a. Penelitian ini hanya membandingkan dua pola arsitektur pengembangan aplikasi, yaitu Model-View-Controller (MVC) dan Model-View-Presenter (MVP). Pola arsitektur lainnya seperti MVVM, VIPER, dan lain-lain tidak termasuk dalam lingkup penelitian ini.
- b. *Maintainability* hanya akan diukur menggunakan metode Maintainability Index dari Code Metrics yang ada dalam dokumentasi Microsoft. Metode selain dari Maintainability Index yang ada dalam Code Metrics Microsoft tidak akan digunakan dalam penelitian ini.
- c. Validasi keakuratan metode Maintainability Index dilakukan dengan mengukur kecepatan peluncuran, kecepatan pengujian, dan konsumsi memori aplikasi berdasarkan studi literatur yang telah dilakukan.
- d. Fokus penelitian ini adalah analisis perbandingan arsitektur MVC dan MVP pada aplikasi iOS menggunakan bahasa pemrograman Swift dengan *framework* UIKit. bahasa pemrograman dan *framework* lain tidak akan dibahas.
- e. Studi kasus dari penelitian ini adalah sebuah aplikasi iOS sederhana yang hanya dapat menampilkan daftar nomor kontak, aplikasi lain atau yang lebih kompleks dari itu tidak akan dibahas.