

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Pengembangan aplikasi menuntut efisiensi dari segi biaya dan waktu [1], [2], [3]. Untuk menjawab tantangan tersebut, banyak pengembang berusaha mengembangkan *tools* atau framework yang ditulis dengan *single code base* yang bisa berjalan diberbagai platform [4], [5], [6]. Pendekatan multiplatform memungkinkan satu basis kode digunakan secara lintas-platform baik mobile, desktop, maupun web, sehingga dapat mengurangi waktu dan biaya pengembangan. Flutter merupakan salah satu framework multi platform terpopuler [7], [8], [9] yang dapat dijadikan pilihan dalam konteks ini. Flutter dipilih dalam penelitian ini karena mendukung pengembangan lintas platform dengan pendekatan *single code base* dan ekosistem yang kuat.

Dalam pengembangan aplikasi multiplatform menggunakan Flutter, penerapan arsitektur yang tepat akan sangat berpengaruh terhadap skalabilitas proyek, kemudahan pengujian, serta fleksibilitas dalam pengembangan dan pemeliharaan aplikasi terutama menangani *dependency* lintas platform [10], [11]. Dua pendekatan arsitektur yang umum digunakan dalam pengembangan perangkat lunak modern adalah Clean Architecture dan Hexagonal Architecture [3]. Keduanya mengusung prinsip pemisahan tanggung jawab (*separation of concerns*) dan modularitas, namun memiliki pendekatan yang berbeda dalam struktur implementasi dan pengelolaan dependensi [1], [12], [13].

Meskipun keduanya memiliki prinsip dasar yang serupa dalam hal pemisahan tanggung jawab, terdapat perbedaan signifikan dalam hal struktur implementasi, cara pengelolaan dependensi, serta pengaruhnya terhadap pengujian dan ekspansi sistem. Sebagian besar studi sebelumnya hanya membahas penerapan arsitektur ini dalam konteks aplikasi tunggal seperti Android, web, atau backend secara terpisah. Penelitian yang secara khusus membandingkan Clean Architecture dan Hexagonal Architecture dalam konteks Flutter untuk

pengembangan aplikasi multiplatform masih sangat terbatas.

Oleh karena itu, penelitian ini bertujuan untuk membandingkan Clean Architecture dan Hexagonal Architecture pada pengembangan aplikasi multiplatform menggunakan Flutter. Penelitian ini diharapkan dapat memberikan wawasan praktis dan teoritis kepada pengembang dalam memilih pendekatan arsitektur yang paling sesuai, serta memberikan kontribusi terhadap praktik pengembangan perangkat lunak yang efisien, terstruktur, dan mudah dikembangkan di masa depan.

### 1.2 Rumusan Masalah

Berdasar latar belakang masalah diatas, rumusan masalah dalam penelitian ini adalah bagaimana perbandingan Clean Architecture dan Hexagonal Architecture terhadap pengelolaan serta pemilihan dependensi eksternal dalam pengembangan aplikasi multiplatform?

### 1.3 Batasan Masalah

Agar penelitian ini terfokus dan terarah, beberapa batasan masalah ditetapkan sebagai berikut:

1. Penelitian ini hanya membahas penerapan Clean Architecture dan Hexagonal Architecture dalam konteks pengembangan aplikasi menggunakan framework Flutter.
2. Platform yang menjadi cakupan dalam pengembangan aplikasi multiplatform terbatas pada Android, Linux (desktop), dan Web. Platform lain seperti iOS, macOS, dan Windows tidak termasuk dalam ruang lingkup penelitian ini.
3. Perbandingan arsitektur dilakukan berdasarkan aspek struktur arsitektur, pengelolaan dependensi, dan kompatibilitas lintas platform. Aspek seperti pengalaman pengembang, dokumentasi, dan preferensi komunitas tidak dianalisis.
4. Evaluasi performa aplikasi dilakukan hanya melalui pengujian *benchmark*

menggunakan Flutter test, dan terbatas pada metrik performa seperti waktu eksekusi fungsi dan respons terhadap operasi dasar. Penelitian tidak mengukur durasi atau efisiensi waktu pengembangan dari masing-masing arsitektur.

5. Studi kasus yang digunakan dalam penelitian ini bersifat representatif dan dibatasi pada aplikasi berskala kecil, dengan fitur-fitur komunikasi dengan API eksternal, dukungan multiplatform, serta kemampuan untuk mengunggah gambar dari penyimpanan lokal.

#### 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Menerapkan Clean Architecture dan Hexagonal Architecture dalam pengembangan aplikasi multiplatform menggunakan framework Flutter.
2. Menganalisis dan membandingkan struktur serta prinsip implementasi antara Clean Architecture dan Hexagonal Architecture dalam konteks Flutter.
3. Mengevaluasi kompatibilitas masing-masing arsitektur terhadap berbagai platform (Android, Linux, dan Web) dalam pengembangan aplikasi multiplatform.
4. Menilai pengaruh arsitektur terhadap pengelolaan dan penggunaan dependensi eksternal dalam proyek Flutter multiplatform.
5. Mengukur performa teknis dari aplikasi yang menggunakan Clean Architecture dan Hexagonal Architecture melalui *benchmark* menggunakan Dart test.
6. Memberikan panduan praktis bagi pengembang dalam memilih arsitektur yang tepat berdasarkan hasil evaluasi untuk meningkatkan kualitas dan produktivitas pengembangan aplikasi multiplatform.

#### 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat Teoritis

- a. Menambah literatur dan wawasan akademik terkait perbandingan penerapan arsitektur perangkat lunak, khususnya Clean Architecture dan Hexagonal Architecture, dalam konteks pengembangan aplikasi multiplatform menggunakan Flutter.
- b. Memberikan dasar analisis yang dapat digunakan untuk penelitian lanjutan mengenai pengaruh arsitektur terhadap performa dan skalabilitas aplikasi lintas platform.

## 2. Manfaat Praktis

- a. Memberikan panduan praktis bagi pengembang dalam memilih arsitektur yang sesuai berdasarkan kebutuhan proyek, terutama untuk aplikasi yang dikembangkan dengan Flutter dan ditargetkan untuk lebih dari satu platform.
- b. Menyediakan data *benchmark* sebagai referensi awal untuk mempertimbangkan efisiensi teknis antara Clean Architecture dan Hexagonal Architecture dalam hal kecepatan eksekusi dan pengelolaan dependensi.

### 1.6 Sistematika Penulisan

Penulisan penelitian ini disusun dalam lima bab utama yang masing-masing memiliki sub bab untuk menjelaskan secara rinci isi penelitian, yaitu:

**BAB I PENDAHULUAN**, Bab ini menjelaskan latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan. Bab ini memberikan gambaran umum mengenai alasan dan arah penelitian.

**BAB II TINJAUAN PUSTAKA**, Bab ini berisi kajian teori yang relevan dengan penelitian, termasuk penjelasan tentang Flutter, Clean Architecture, Hexagonal Architecture, serta penelitian-penelitian sebelumnya yang mendukung.

**BAB III METODE PENELITIAN**, Bab ini menjelaskan metode yang digunakan dalam penelitian, termasuk jenis dan pendekatan penelitian, desain dan

objek penelitian, tahapan implementasi, teknik pengumpulan data, serta metode analisis perbandingan dan pengujian benchmark.

**BAB IV HASIL DAN PEMBAHASAN**, Bab ini menyajikan hasil implementasi aplikasi menggunakan kedua arsitektur, hasil benchmark performa, serta analisis perbandingan dari berbagai aspek seperti struktur arsitektur, pengelolaan dependensi, dan kompatibilitas lintas platform. Pembahasan dilakukan secara kritis berdasarkan temuan empiris.

**BAB V PENUTUP**, Bab ini memuat kesimpulan dari hasil penelitian dan saran yang dapat dijadikan acuan untuk pengembangan atau penelitian lebih lanjut di masa depan.

