PERBANDINGAN CLEAN ARCHITECTURE DAN HEXAGONAL ARCHITECTURE PADA PENGEMBANGAN APLIKASI MULTIPLATFORM MENGGUNAKAN FRAMEWORK FLUTTER

SKRIPSI

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana

Program Studi Informatika

disusun oleh

Nofal Briansah

18.11.2253

Kepada

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

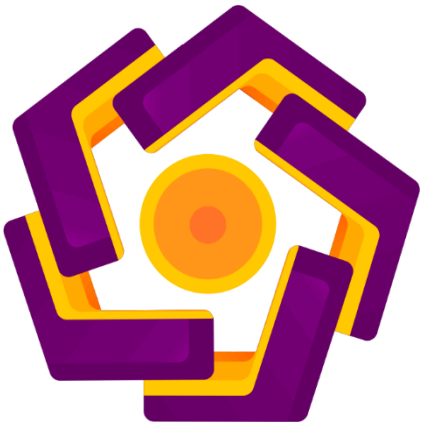
2025

PERBANDINGAN CLEAN ARCHITECTURE DAN HEXAGONAL ARCHITECTURE PADA PENGEMBANGAN APLIKASI MULTIPLATFORM MENGGUNAKAN FRAMEWORK FLUTTER

SKRIPSI

untuk memenuhi salah satu syarat mencapai derajat Sarjana

Program Studi Informatika



disusun oleh

Nofal Briansah

18.11.2253

Kepada

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2025

HALAMAN PERSETUJUAN

SKRIPSI

PERBANDINGAN CLEAN ARCHITECTURE DAN HEXAGONAL ARCHITECTURE PADA PENGEMBANGAN APLIKASI MULTIPLATFORM MENGGUNAKAN FRAMEWORK FLUTTER

yang disusun dan diajukan oleh

Nofal Briansah

18.11.2253

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 10 Juli 2025

Dosen Pembimbing,

Agit Amrullah, M.Kom
NIK. 190302356

HALAMAN PENGESAHAN

SKRIPSI

PERBANDINGAN CLEAN ARCHITECTURE DAN HEXAGONAL ARCHITECTURE PADA PENGEMBANGAN APLIKASI MULTIPLATFORM MENGGUNAKAN FRAMEWORK FLUTTER

yang disusun dan diajukan oleh

Nofal Briansah

18.11.2253

Telah dipertahankan di depan Dewan Penguji
pada tanggal 30 Juli 2025

Susunan Dewan Penguji

Nama Penguji	Tanda Tangan
Rifda Faticha Alfa Aziza, M.Kom **NIK. 190302392**	
Ainul Yaqin, M.Kom **NIK. 190302255**	
Subektiningsih, M.Kom **NIK. 190302413**	

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 30 Juli 2025

DEKAN FAKULTAS ILMU KOMPUTER

Prof. Dr. Kusrini, S.Kom., M.Kom.
NIK. 19030209

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Nofal Briansah
NIM : 18.11.2253

Menyatakan bahwa Skripsi dengan judul berikut:

Perbandingan Clean Architecture Dan Hexagonal Architecture Pada Pengembangan Aplikasi Multiplatform Menggunakan Framework Flutter

Dosen Pembimbing : Agit Amrullah, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 10 Juli 2025

Yang Menyatakan,

Nofal Briansa

HALAMAN PERSEMBAHAN

Dengan rasa syukur dan hormat yang mendalam, Skripsi ini Penulis persembahkan kepada:

1. Kedua orang tua, Bapak Rokani dan Ibu Dartem, atas segala dukungan, kasih sayang, dan pengorbanan dalam setiap langkah perjalanan hidup penulis.
2. Mbah kalem, dengan kasih sayang tulus serta terima kasih atas segala hal yang telah diberikan.
3. Kakak Ari Susanti, dengan hormat dan terima kasih yang mendalam atas segala kebaikan, perhatian, serta masukan yang telah diberikan kepada penulis
4. Adik-adik: Ratna, Winda, dan Chiel, yang senantiasa memberikan semangat dan dukungan.
5. Bapak Agit Amrullah, M.Kom, Ibu Mardhiya Hayaty, S.T., M.Kom, dan Ibu Eli Pujastuti, M.Kom. Selaku Dosen Pembimbing, Dosen Wali, dan Kaprodi Informatika, serta seluruh dosen pengajar yang telah memberikan ilmu, bimbingan, dan arahan yang sangat berharga selama masa perkuliahan.
6. Rekan-rekan dan sahabat seperjuangan Informatika 7, yang telah menjadi bagian penting dalam perjalanan akademik ini melalui kebersamaan, dukungan, dan inspirasi.
7. Teman-teman Trans4mers, yang turut memberikan warna dan makna kepada penulis.
8. Kepada Christopher Tin, Sid Meier's Civilization VI, dan seluruh tim kreatif di baliknya, yang telah menghidupkan lagu *Sogno di Volare* sebuah karya luar biasa yang telah memberikan semangat dan inspirasi.
9. Untuk Rainych Ran, yang lewat suaranya telah menjadi sumber semangat dan ketenangan sepanjang proses ini. Terima kasih telah menemani dengan harmoni yang tulus.

KATA PENGANTAR

Dengan penuh rasa syukur kepada Tuhan atas kemudahan dan kesempatan yang diberikan, penulis berhasil menyelesaikan skripsi yang berjudul "Perbandingan Clean Architecture dan Hexagonal Architecture pada Pengembangan Aplikasi Multiplatform Menggunakan Framework Flutter".

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika. Dalam proses penyusunannya, penulis mendapatkan banyak bantuan, arahan, dan dukungan dari berbagai pihak, yang dengan tulus penulis ucapkan terima kasih, kepada:

1. Bapak Agit Amrullah, M.Kom selaku Dosen Pembimbing.
2. Ibu Mardhiya Hayaty, S.T., M.Kom selaku Dosen Wali.
3. Ibu Eli Pujastuti, M.Kom selaku Kaprodi Informatika.
4. Segenap Dosen dan Staff Universitas Amikom Yogyakarta.
5. Orang tua dan keluarga.
6. Rekan-rekan Informatika 7.
7. Teman-teman Trans4mers.

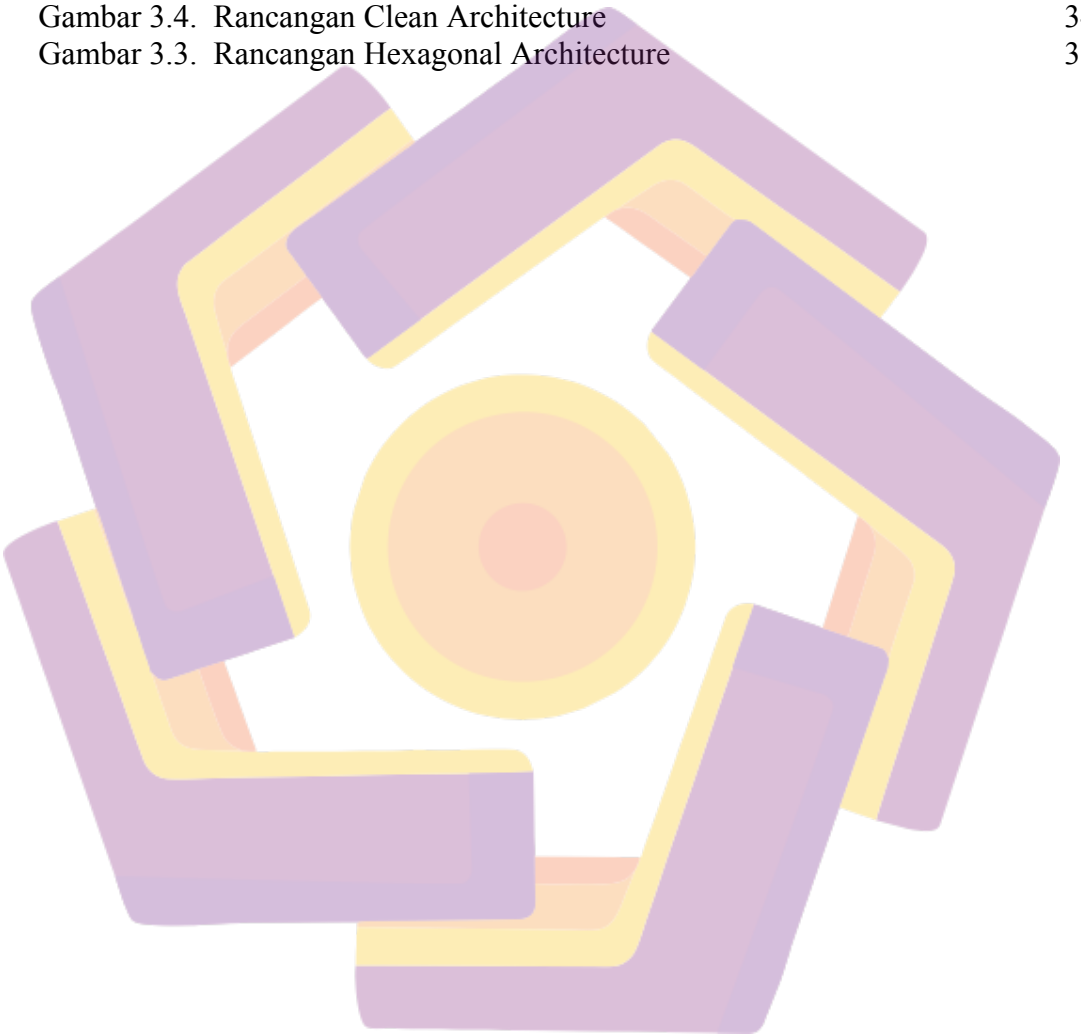
Penulis menyadari bahwa skripsi ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan. Semoga karya ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya dalam pengembangan aplikasi multiplatform.

Yogyakarta, 10 Juli 2025
Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR



DAFTAR LAMPIRAN

DAFTAR ISTILAH

Clean Architecture	Pola arsitektur perangkat lunak yang memisahkan kode menjadi beberapa lapisan untuk meningkatkan modularitas dan *maintainability*.
Hexagonal Architecture	Pola arsitektur perangkat lunak yang menggunakan port dan adapter untuk mengisolasi logika bisnis inti dari antarmuka pengguna dan infrastruktur.
Flutter	Framework open-source dari Google untuk membangun aplikasi multiplatform menggunakan bahasa Dart.
Multiplatform	Kemampuan aplikasi untuk berjalan pada berbagai platform seperti Android, iOS, Web, dan Desktop.
Framework	Kerangka kerja perangkat lunak yang menyediakan fondasi dan alat untuk pengembangan aplikasi.
Lapisan (Layer)	Bagian terpisah dalam arsitektur perangkat lunak dengan fungsi tertentu seperti presentasi, domain, dan data.
Port dan Adapter	Konsep dalam Hexagonal Architecture yang menghubungkan aplikasi dengan dunia luar melalui antarmuka dan implementasi konkret.
Modularitas	Prinsip desain yang membagi sistem menjadi modul-modul kecil yang dapat dikembangkan dan diuji secara independen.
Testabilitas	Kemudahan dalam melakukan pengujian pada kode atau sistem perangkat lunak. Berdasarkan testing yang telah dibuat

Maintainabilitas	Kemudahan dalam memelihara dan mengembangkan aplikasi setelah penerapan.
Dart	Bahasa pemrograman yang digunakan dalam pengembangan aplikasi Flutter.
Dependency Injection	Teknik yang memungkinkan objek menerima dependensi dari luar untuk meningkatkan fleksibilitas dan testabilitas.
Use Case	Representasi fungsi atau aksi aplikasi untuk memenuhi kebutuhan pengguna.
Entity	Objek domain yang berisi aturan bisnis dan data inti dalam Clean Architecture

INTISARI

Penelitian ini bertujuan untuk membandingkan penerapan Clean Architecture dan Hexagonal Architecture dalam pengembangan aplikasi multiplatform berbasis Flutter, dengan fokus pada efisiensi pengujian dan pengelolaan dependensi eksternal. Metode yang digunakan adalah analisis komparatif eksperimental, di mana dua aplikasi dengan fitur identik yaitu percakapan AI berbasis Google Gemini API dibangun dan diuji pada tiga platform, Android, Linux, dan Web.

Evaluasi dilakukan berdasarkan tiga aspek utama, struktur arsitektur, pengelolaan dependensi eksternal, dan performa unit test. Hasil menunjukkan bahwa kedua pendekatan arsitektur mendukung prinsip modularitas, separation of concerns, dan testability. Pada aspek performa, Clean Architecture mencatat rata-rata waktu eksekusi unit test sebesar 35,45 ms, sedikit lebih efisien dibandingkan Hexagonal Architecture dengan 41,17 ms.

Perbedaan signifikan terlihat pada pengelolaan dependensi eksternal. Clean Architecture menggunakan pendekatan pemisahan lapisan tanpa kontrak eksplisit terhadap dependensi eksternal, yang sederhana namun kurang fleksibel terhadap perubahan teknologi. Sebaliknya, Hexagonal Architecture menerapkan pola ports and adapters secara eksplisit, yang memungkinkan pengujian lebih fleksibel dan penggantian dependensi eksternal tanpa mengubah logika inti.

Hasil penelitian ini menyimpulkan bahwa pemilihan arsitektur sangat bergantung pada konteks proyek. Clean Architecture cocok untuk sistem yang sederhana dan tim kecil, sementara Hexagonal Architecture lebih tepat untuk sistem kompleks yang membutuhkan fleksibilitas dan skalabilitas jangka panjang.

Kata kunci: Clean Architecture, Hexagonal Architecture, Flutter, Multiplatform, Arsitektur Perangkat Lunak

ABSTRACT

This study aims to compare the implementation of Clean Architecture and Hexagonal Architecture in the development of multiplatform applications using Flutter, with a focus on test efficiency and the management of external dependencies. The research employs a comparative experimental analysis method, where two applications with identical features an AI-based chat using the Google Gemini API were developed and tested on Android, Linux, and Web platforms.

The evaluation is based on three main aspects, architectural structure, external dependency management, and unit test performance. The results show that both architectural approaches support the principles of modularity, separation of concerns, and testability. In terms of unit test performance, Clean Architecture recorded an average execution time of 35.45 ms, slightly faster than Hexagonal Architecture, which averaged 41.17 ms.

The key difference lies in the management of external dependencies. Clean Architecture applies a layered separation approach without explicit contracts for external dependencies, which is simpler but less adaptable to technological changes. In contrast, Hexagonal Architecture utilizes the ports and adapters pattern, enabling more flexible testing and easier replacement of external technologies without altering core logic.

The findings suggest that architectural selection should be aligned with the project context. Clean Architecture is more suitable for simpler systems with smaller teams, while Hexagonal Architecture is ideal for complex systems requiring flexibility and long-term scalability.

***Keyword*:** Clean Architecture, Hexagonal Architecture, Flutter, Multiplatform, Software Architecture