ANALISIS PENERAPAN CLEAN ARCHITECTURE TERHADAP MAINTAINABILITY DAN COMPLEXITY KODE PADA PENGEMBANGAN APLIKASI MOBILE BERBASIS FLUTTER

SKRIPSI

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Informatika

disusun oleh

Allan Jati Prakoso

19.11.2666

Kepada

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

ANALISIS PENERAPAN CLEAN ARCHITECTURE TERHADAP MAINTAINABILITY DAN COMPLEXITY KODE PADA PENGEMBANGAN APLIKASI MOBILE BERBASIS FLUTTER

SKRIPSI

untuk memenuhi salah satu syarat mencapai derajat Sarjana

Program Studi Informatika

disusun oleh

Allan Jati Prakoso

19.11.2666

Kepada

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

HALAMAN PERSETUJUAN

SKRIPSI

ANALISIS PENERAPAN CLEAN ARCHITECTURE TERHADAP MAINTAINABILITY DAN COMPLEXITY KODE PADA PENGEMBANGAN APLIKASI MOBILE BERBASIS FLUTTER

yang disusun dan diajukan oleh

Allan Jati Prakoso
19.11.2666

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 5 July 2023

Dosen Pembimbing,

Arif Akbarul Huda, S.Si.,M.Eng.
NIK. 1903002287

HALAMAN PENGESAHAN

SKRIPSI

ANALISIS PENERAPAN CLEAN ARCHITECTURE TERHADAP MAINTAINABILITY DAN COMPLEXITY KODE PADA PENGEMBANGAN APLIKASI MOBILE BERBASIS FLUTTER

yang disusun dan diajukan oleh

Allan Jati Prakoso
19.11.2666

Telah dipertahankan di depan Dewan Penguji
pada tanggal 20 Juli 2023

Susunan Dewan Penguji

Nama Penguji	Tanda Tangan
Arif Akbarul Huda, S.Si, M.Eng **NIK. 190302287**	
Muhammad Koprawi, S.Kom., M.Eng **NIK. 190302454**	
Jeki Kuswanto, M.Kom **NIK. 190302456**	

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 20 Juli 2023

DEKAN FAKULTAS ILMU KOMPUTER

Hanif Al Fatta,S.Kom., M.Kom.
NIK. 190302096

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Allan Jati Prakoso
NIM : 19.11.2666

Menyatakan bahwa Skripsi dengan judul berikut:

Analisis Penerapan Clean Architecture Terhadap Maintainability Dan Complexity Kode Pada Pengembangan Aplikasi Mobile Berbasis Flutter

Dosen Pembimbing : Arif Akbarul Huda, S.Si, M.Eng

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, 20 July 2023

Yang Menyatakan,

Allan Jati Prakoso

KATA PENGANTAR

Puji syukur kehadirat Tuhan yang maha esa, yang telah memberikan rahmat, hidayah, serta kekuatan sehingga penulis dapat menyelesaikan skripsi ini tepat pada waktunya. Penyelesaian skripsi ini tidak terlepas dari bantuan, dukungan, serta motivasi dari berbagai pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung.

Oleh karena itu, melalui kesempatan ini, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Arif Akbarul Huda, S.si, M.Eng, sebagai Dosen Pembimbing yang telah memberikan arahan, masukan, kritik, serta saran dalam proses penulisan skripsi ini.
2. Ibu dan Kakak, yang dengan tulus memberikan dukungan moril dan materil, doa, serta semangat yang tak pernah putus selama proses penyusunan skripsi ini.
3. Seluruh teman-teman penulis, Terutama Caroline Sentiaji, yang telah memberikan dukungan, motivasi, dan semangat selama periode penulisan skripsi.
4. Seluruh pihak yang telah memberikan dukungan baik dalam bentuk data, saran, maupun motivasi, sehingga skripsi ini dapat terselesaikan dengan baik.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Kritik dan saran dari semua pihak sangat diharapkan guna penyempurnaan skripsi ini. Semoga hasil dari penulisan skripsi ini dapat memberikan manfaat, khususnya bagi penulis dan umumnya bagi pihak-pihak yang berkepentingan.

Yogyakarta, 20 Juli 2023

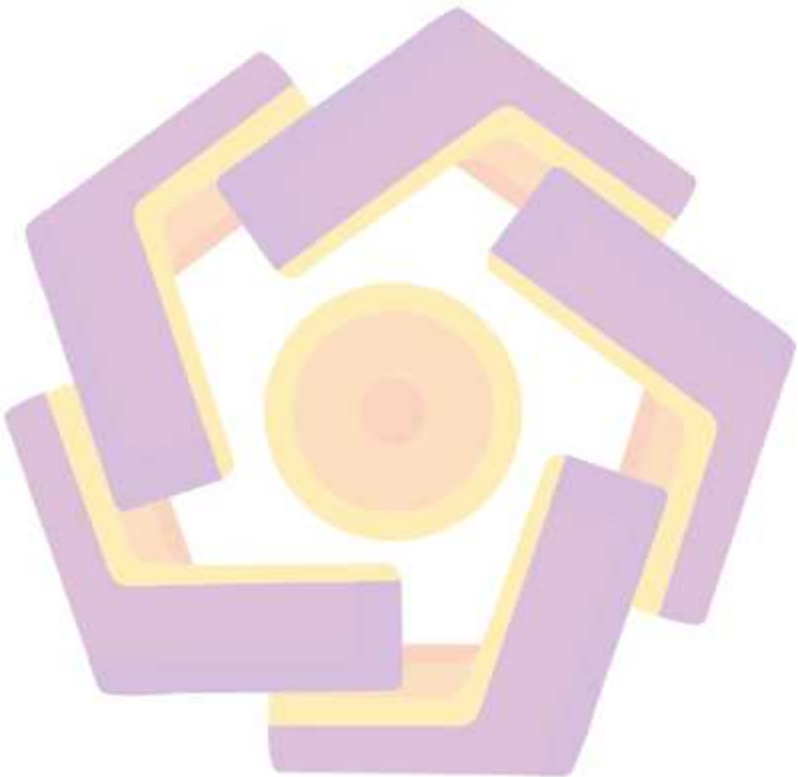
Penulis

DAFTAR ISI

DAFTAR TABEL

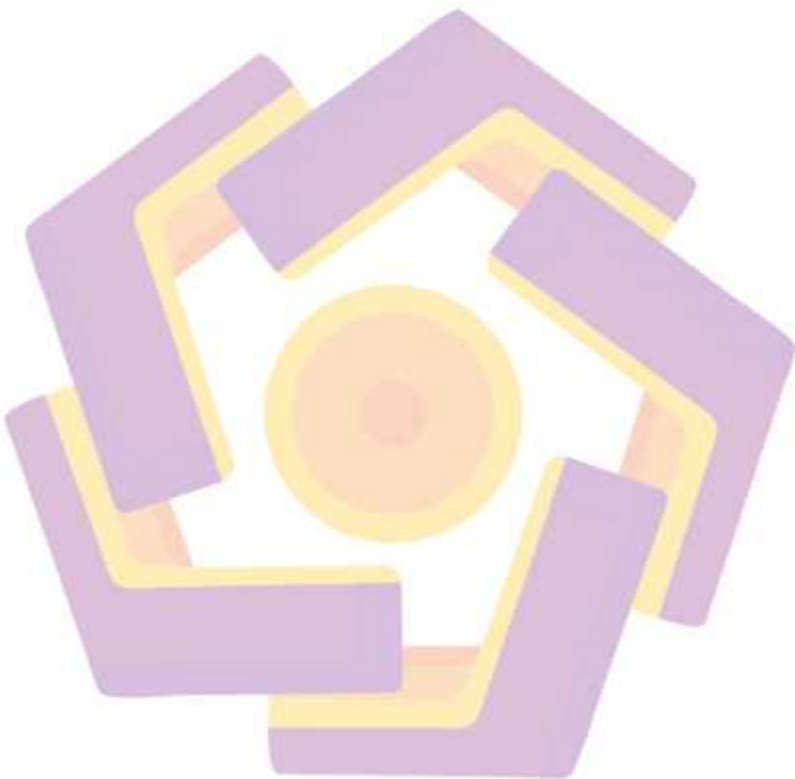
DAFTAR GAMBAR

DAFTAR LAMPIRAN



DAFTAR LAMBANG DAN SINGKATAN

HV	Halstead Metrics Volumes
CC	Cyclomatic Complexity
LOC	Lines of Code
PerCm	Percent line of comment



DAFTAR ISTILAH

Clean Architecture	Arsitektur perangkat lunak yang berfokus pada pemisahan logika bisnis inti dari infrastruktur dan perangkat lunak luar.
Maintainability	Kemampuan suatu perangkat lunak untuk diubah, diperbaiki, dan diperbarui dengan mudah.
Complexity	Tingkat kesulitan atau kompleksitas suatu perangkat lunak, yang dapat dipengaruhi oleh jumlah fitur, ketergantungan, dan struktur kode.
Flutter	Framework open-source untuk pengembangan aplikasi mobile yang menggunakan bahasa pemrograman Dart.
Dart	Bahasa pemrograman yang digunakan untuk mengembangkan aplikasi Flutter.
Model	Bagian dari pola arsitektur perangkat lunak yang bertanggung jawab untuk memodelkan data dan logika bisnis.
View	Bagian dari pola arsitektur perangkat lunak yang berfungsi untuk menampilkan tampilan dan menerima input dari pengguna.
State	Keadaan atau kondisi dari suatu komponen dalam aplikasi, yang dapat berubah seiring interaksi atau perubahan data.
Widget	Komponen UI dalam Flutter yang dapat dibangun dan dikombinasikan untuk membentuk antarmuka pengguna.
Context	Objek yang memberikan akses ke berbagai fitur dan fungsi dalam framework Flutter.
Consumer	Widget dalam Flutter yang mengonsumsi data dari Provider dan mengubah tampilan berdasarkan perubahan data.
Controller	Bagian dari pola arsitektur perangkat lunak yang bertanggung jawab untuk mengatur logika bisnis dan berinteraksi dengan model dan view.
Presenter	Bagian dari pola arsitektur perangkat lunak yang bertanggung jawab untuk mengatur tampilan dan menangani interaksi pengguna.

Entities	Representasi objek yang memiliki atribut dan perilaku dalam konteks domain aplikasi.
Use Cases	Fungsi-fungsi atau proses bisnis yang diimplementasikan dalam aplikasi untuk mencapai tujuan tertentu.
Interface	Kontrak atau perjanjian yang mendefinisikan metode dan perilaku yang harus diimplementasikan oleh kelas-kelas tertentu.
Adapters	Komponen yang digunakan untuk menghubungkan antara bagian-bagian yang berbeda dalam sistem perangkat lunak.
Dependency Injection	Teknik dalam pengembangan perangkat lunak untuk memasukkan dependensi ke dalam komponen secara terpisah, memungkinkan penggantian dan pengujian yang lebih mudah.
Provider	Library di Flutter untuk mengelola state aplikasi dan berbagi data antara komponen-komponen.
State Management	Pendekatan dan teknik untuk mengelola dan memperbarui keadaan aplikasi dalam pengembangan perangkat lunak.
Cyclomatic Complexity	Ukuran yang mengukur kompleksitas kontrol alur suatu program berdasarkan jumlah jalur yang berbeda.
Maintainability Index	Indeks yang mengukur tingkat maintainability suatu perangkat lunak berdasarkan faktor-faktor seperti kompleksitas, ukuran, dan dokumentasi.
Design Pattern	Solusi umum yang telah terbukti dalam merancang dan mengembangkan perangkat lunak untuk menyelesaikan masalah yang sering muncul.

INTISARI

Revolusi industri 4.0 menimbulkan tantangan bagi pengembang aplikasi *mobile* untuk menciptakan produk yang responsif dan adaptif terhadap perubahan. Salah satu pendekatan yang dapat dilakukan adalah dengan meningkatkan *maintainability* dan *code complexity*. Penelitian ini mengevaluasi penerapan *clean architecture* dalam pengembangan aplikasi *mobile* berbasis *Flutter*, khususnya aplikasi *Intention Habit* milik peneliti yang akan dijadikan baseline penelitian ini. *Flutter* digunakan karena kemampuannya menghasilkan aplikasi multi-platform dan fitur Hot Reload yang mempercepat proses pengembangan.

Metode penelitian ini mencakup studi pustaka, perencanaan, perancangan aplikasi, implementasi sistem, evaluasi *maintainability* dan *complexity*, serta analisis *data* hasil evaluasi. Evaluasi menggunakan *Dart Code* Metrics untuk mengukur kualitas kode aplikasi *Intention Habit*. Setelah itu, analisis *data* dilakukan untuk mengeksplorasi dan menginterpretasikan temuan serta mengkategorikan tingkat *maintainability* dan *complexity*.

Hasil penelitian menunjukkan nilai *maintainability* index 78 dan *cyclomatic complexity* 569, yang menegaskan pentingnya struktur aplikasi yang baik dan modular. Penelitian ini mengidentifikasi dan menganalisis tingkat *maintainability* dan *complexity* serta menggali area potensial perbaikan. Temuan ini dapat dimanfaatkan oleh pengembang aplikasi dan pemangku kepentingan proyek pengembangan aplikasi untuk meningkatkan efisiensi dan efektivitas pengembangan aplikasi berbasis *Flutter* dan *Dart*, serta memberikan wawasan tentang optimasi arsitektur aplikasi yang lebih baik.

Kata kunci: *Flutter, Dart, Clean architecture, Maintainability, Complexity*

ABSTRACT

The industry 4.0 revolution poses challenges for mobile application developers to create products that are responsive and adaptive to change. One approach that can be taken is to improve maintainability and code complexity. This study evaluates the implementation of clean architecture in the development of Flutter-based mobile applications, specifically the Intention Habit application owned by the researcher, which will serve as the baseline for this study. Flutter is used for its ability to produce multi-platform applications and its Hot Reload feature, which accelerates the development process.

The research method includes literature review, planning, application design, system implementation, maintainability and complexity evaluation, and analysis of evaluation data results. The evaluation uses Dart Code Metrics to measure the code quality of the Intention Habit application. Following this, data analysis is conducted to explore and interpret findings and categorize the levels of maintainability and complexity.

The research results show a maintainability index of 78 and a cyclomatic complexity of 569, emphasizing the importance of a well-structured and modular application. This study identifies and analyzes the levels of maintainability and complexity, as well as uncovering potential areas for improvement. These findings can be utilized by application developers and stakeholders in application development projects to enhance efficiency and effectiveness in developing Flutter and Dart-based applications and provide insights for better application architecture optimization.

Keywords: *Flutter, Dart, Clean Architecture, Maintainability, Complexity*