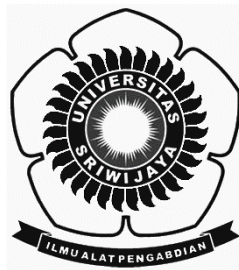


**THE APPLICATION OF CLEAN ARCHITECTURE PRINCIPLES
WITH MVVM (MODEL-VIEW-VIEWMODEL), REACTIVE
PROGRAMMING, AND DECLARATIVE UI (USER INTERFACE) IN
MODERN ANDROID DEVELOPMENT**

Diajukan Sebagai Syarat Untuk Menyelesaikan
Pendidikan Program Strata-1 Pada
Jurusan Teknik Informatika



Oleh:

Mahmoud Darwish
NIM: 09021081924166

Jurusan Teknik Informatika
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA

2023

LEMBAR PENGESAHAN SKRIPSI

THE APPLICATION OF CLEAN ARCHITECTURE PRINCIPLES WITH MVVM (MODEL-VIEW-VIEWMODEL), REACTIVE PROGRAMMING, AND DECLARATIVE UI (USER INTERFACE) IN MODERN ANDROID DEVELOPMENT

Oleh:

Mahmoud Darwish

NIM: 09021081924166

Palembang, 11/Jul/2023

Pembimbing I



Julian Supardi, M.T.

NIP.197207102010121001

Pembimbing II



Hadipurnawan satria, Ph.D

NIP.198004182020121001

Mengetahui,

Ketua Jurusan Teknik Informatika



Aly Syahrini Utami, M.Kom.

NIP.197812222006042003

TANDA LULUS UJIAN KOMPREHENSIF

Pada hari Jumat tanggal 7 Juli 2023 telah dilaksanakan ujian komprehensif skripsi oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya

Nama : Mahmoud Darwish
NIM : 09021081924166
Judul : The Application of Clean Architecture Principles with MVVM
(Model-View-ViewModel), Reactive Programming, and Declarative UI
(User Interface) in Modern Android Development

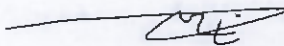
dan dinyatakan LULUS.

Ketua Penguji



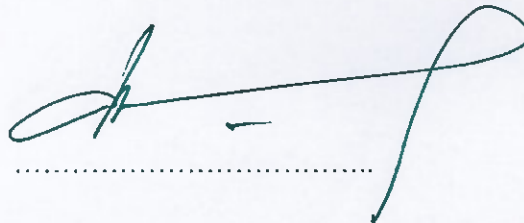
Alvi Syahrini Utami, M.Kom.
NIP. 19781222200642003

Penguji I



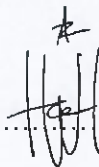
Osvari Arsalan, S.Kom., M.T.
NIP. 198806282018031001

Pembimbing I



Julian Supardi, M.T.
NIP.197207102010121001

Pembimbing II



Hadipurnawan satria, Ph.D
NIP.198004182020121001

Mengetahui,
Ketua Jurusan Teknik Informatika



Alvi Syahrini Utami, M.Kom.
NIP. 19781222200642003

HALAMAN MOTTO DAN PERSEMBAHAN

Motto:

اللهم إني أنزلت بك حاجاتي كلها الظاهرة والباطنة، الدنيوية والأخروية، عبديك بفنائك، مسكينك
بفنائك فقيرك بفنائك، يا من لا يعلم كيف هو إلا هو، ويا من لا يبلغ قدرته عجزه، يا شاهداً غير غائب، ويا
قريباً غير بعيد، ويا غالباً غير مغلوب، يا حي يا قيوم، بحولك وقوتك أستعين وأستجير فارحمني يا أرحم
الرحمين

Kupersembahkan karya tulis ini kepada:

Allah glorified be He

The best of creation the prophet of Allah Muhammad peace and blessings be
forever be on him, his blessed family, his companions, and everyone who
accepts him as Allah's final messenger

ABSTRACT

Clean Architecture principles, Model-View-ViewModel (MVVM) architecture, Reactive Programming, and declarative UI (user interface) are a set of principles and practices aimed at producing highly maintainable, scalable, and testable applications. In this thesis, the writer demonstrates how the application of those principles and practices when implemented can produce maintainable, scalable, and testable applications and supports that with tests all of which have passed successfully. The application produced is for an institution in Indonesia that was tracking the absent students from the daily meetings using semi-manual methods that were consuming more time and energy than they had hoped. The application helped them reduce the time needed to create new reports and helped them entirely automate processing the reports.

Keywords: Clean Architecture, MVVM, Reactive Programming, Declarative UI, Android Development, Absence Tracking.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji dan syukur kehadiran Allah SWT atas berkat rahmat dan ridho yang diberikan kepada peneliti sehingga dapat menyelesaikan Tugas Akhir berupa Skripsi yang disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan program Strata-1 pada Fakultas Ilmu Komputer Program Studi Teknik Informatika Universitas Sriwijaya.

Dalam pengerjaan Skripsi ini banyak kerabat dan guru yang telah memberikan bantuan serta bimbingan baik secara akademis dan non-akademis. Pada kesempatan kali ini, penulis akan menyampaikan ucapan terimakasih kepada pihak-pihak yang telah membantu penulis, yaitu:

1. Allah, glorified be He, without whose care and love I would have not even been.
2. The best of creation the last prophet of Allah Muhammad, may peace and blessing forever be on him, his family, companions, and everyone who follows his guidance.
3. My mother and father for their love and prayers.
4. Bapak Jaidan Jauhari, S.Pd., M.T. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Ibu Alvi Syahrini Utami, M.Kom. selaku ketua Jurusan Teknik Informatika serta menjadi Ketua Penguji dalam sidang komprehensif.
6. Bapak Julian Supardi, M.T. selaku pembimbing tugas akhir.
7. Bapak Hadipurnawan satria, Ph.D selaku pembimbing tugas akhir.
8. Bapak Osvari Arsalan, S.Kom, M.T selaku Penguji tugas akhir.

Dalam penulisan Skripsi ini penulis menyadari bahwa masih banyak kekurangan dikarenakan keterbatasan pemikiran, pengetahuan, serta pengalaman, maka dari itu diperlukan kritik dan saran untuk kedepannya.

Palembang, 11 Juli 2023

Mahmoud Darwish

HALAMAN PERNYATAAN

Yang bertanda tangan dan membuat pernyataan ini:

Nama : Mahmoud Darwish
NIM : 09021081924166
Program Studi : Teknik Informatika Bilingual
Judul Skripsi : The Application of Clean Architecture Principles with MVVM (Model-View-ViewModel), Reactive Programming, and Declarative UI (User Interface) in Modern Android Development
Hasil Pengecekan : 6%
Plagiat Software
iThenticate/Turnitin

Menyatakan bahwa dokumen Skripsi dari hasil penelitian adalah hasil karya milik saya sendiri bukan hasil jiplakan/plagiat dari pihak manapun. Apabila ditemukan unsur penjiplakan/plagiat melebihi dari persentase yang berlaku di Fakultas Ilmu Komputer Universitas Sriwijaya (20% Kemiripan), maka saya bersedia dikenakan sanksi akademik dari Universitas Sriwijaya.

Demikian pernyataan ini saya buat dengan jujur dan sebenar-benarnya tanpa ada tekanan/paksaan dari pihak manapun.



Palembang, 11 Juli 2023



Mahmoud Darwish
NIM: 09021081924166

DAFTAR ISI

Halaman

LEMBAR PENGESAHAN SKRIPSI.....	ii
TANDA LULUS UJIAN KOMPREHENSIF	iii
HALAMAN MOTTO DAN PERSEMBAHAN.....	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
HALAMAN PERNYATAAN.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
Chapter I INTRODUCTION	I-1
I.1 Introduction.....	I-1
I.2 Research Problem	I-2
I.3 Research Aims	I-2
I.4 Research Benefits.....	I-2
I.5 Project Scope	I-3
I.6 Writing Organizational System.....	I-3
I.6.1 Chapter 1	I-3
I.6.2 Chapter 2.....	I-3
I.6.3 Chapter 3	I-3
I.6.4 Chapter 4.....	I-3
I.6.5 Chapter 5	I-4

I.6.6 Chapter 6	I-4
I.7 Summary	I-4

Chapter II FOUNDATIONAL CONCEPTS AND RELEVANT RESEARCH II-1

II.1 Introduction	II-1
II.2 Foundational Concepts	II-1
II.2.1 Software architecture	II-1
II.2.2 Android	II-1
II.2.3 Model-View-ViewModel (MVVM)	II-1
II.2.4 Clean Architecture and Reactive Programming	II-3
II.2.5 Imperative UI	II-6
II.2.6 Declarative UI	II-6
II.2.7 Modern Android Development	II-7
II.2.8 Android Studio	II-7
II.2.9 Version Control Systems (VCS) and Git	II-7
II.2.10 Agile Methodology	II-8
II.3 Related publications	II-8
II.3.1 Alexandru Dumbravan – Clean Android Architecture – Take a layered approach to writing clean, testable, and decoupled Android applications – Packt Publishing (2022)	II-8
II.3.2 Alexey Soshin – Kotlin Design Patterns and Best Practices – Build scalable applications using traditional, reactive, and concurrent design patterns in Kotlin, 2 nd Edition – Packt Publishing (2022)	II-9
II.3.3 Thomas Künneth – Android UI Development with Jetpack Compose – Packt Publishing (2022)	II-9
II.4 Summary	II-10

Chapter III RESEARCH METHODOLOGY..... III-1

III.1 Introduction	III-1
III.2 Project stakeholders.....	III-1
III.3 Data Collection.....	III-2
III.3.1 Work Framework	III-2
III.3.2 Data Type	III-2
III.3.3 Data Source	III-2
III.3.4 Data Collection Method	III-2
III.4 Research Framework.....	III-3
III.4.1 The Scope and Stakeholders	III-3
III.4.2 Foundational Concepts and Relevant Publications	III-3
III.4.3 The Testing Mechanism	III-3
III.4.4 The Tools to Be Used.....	III-3
III.4.5 Testing Data Format.....	III-4
III.4.6 Testing Prosses	III-4
III.5 Testing process	III-4
III.6 Application Development Method	III-5
III.6.1 The Requirements Collection Phase	III-5
III.6.2 The Design Phase	III-6
III.6.3 The Implementation Phase	III-6
III.6.4 The Testing Phase	III-6
III.6.5 The Distribution Phase	III-6
III.7 Project Management.....	III-6
III.8 Summary	III-7

Chapter IV APPLICATION DEVELOPMENT PROCESSIV-8

IV.1 Introduction.....	IV-8
IV.2 Analysis.....	IV-8
IV.2.1 Use cases	IV-8
IV.2.2 Class diagrams	IV-22

IV.2.3 Sequence diagrams.....	IV-31
IV.2.4 ER diagram	IV-36
IV.3 Design	IV-37
IV.3.1 Unauthenticated	IV-37
IV.3.2 Authenticated student home.....	IV-37
IV.3.3 Submission history	IV-37
IV.3.4 Authenticated supervisor home.....	IV-37
IV.3.5 Reports lists.....	IV-38
IV.3.6 Report editing.....	IV-39
IV.3.7 Authenticated admin home	IV-41
IV.3.8 Statistics	IV-42
IV.3.9 User management.....	IV-42
IV.3.10 Report submission.....	IV-42
IV.4 Implementation	IV-44
IV.4.1 Data storage.....	IV-44
IV.4.2 Code organization	IV-44
IV.4.3 Essential concepts	IV-45
IV.4.4 Implemented user interface designs	IV-52
IV.5 Testing.....	IV-59
IV.5.1 Unit tests	IV-60
IV.5.2 End-to-end instrumental tests.....	IV-63
IV.6 Release	IV-65
IV.7 Summary	IV-67
Chapter V RESULTS AND ANALYSIS.....	V-1
V.1 Unit tests results	V-1
V.2 End-to-end instrumental tests.....	V-3
V.3 Summary	V-4

Chapter VI CONCLUSIONS AND SUGGESTIONS.....	VI-1
VI.1 Conclusions	VI-1
VI.2 Suggestions	VI-1
DAFTAR PUSTAKA	iii
LAMPIRAN.....	v
Source Code Repository	v
AppTester Invitation Link	v

DAFTAR TABEL

Table IV-1 Actors.....	IV-11
Table IV-2 Student use cases	IV-12
Table IV-3 Supervisor use cases	IV-12
Table IV-4 Admin use cases	IV-13
Table IV-5 Login use case.....	IV-14
Table IV-6 Submission history use case	IV-15
Table IV-7 Report submission use case	IV-16
Table IV-8 Statistics use case.....	IV-17
Table IV-9 Assigned groups reports use case	IV-18
Table IV-10 Deleting reports	IV-20
Table IV-11 User permissions management use case	IV-21
Table IV-12 Implemented classes	IV-46
Table IV-13 Unit tests descriptions.....	IV-60
Table IV-14 End-to-end tests descriptions.....	IV-63
Table V-1 Unit tests results	V-1
Table V-2 Instrumental end-to-end tests results	V-3

DAFTAR GAMBAR

Figure II-1 MVVM architecture overview	II-2
Figure II-2 MVVM data flow overview	II-3
Figure II-3 Clean Architecture overview	II-4
Figure II-4 Clean Architecture dependency structure	II-5
Figure III-1 Work Framework Overview	III-2
Figure IV-1 Use cases overview	IV-9
Figure IV-2 Student use cases	IV-10
Figure IV-3 Manager use cases	IV-10
Figure IV-4 Admin use cases	IV-11
Figure IV-5 Main class diagram.....	IV-22
Figure IV-6 Authentication class diagram	IV-23
Figure IV-7 Submission history class diagram	IV-24
Figure IV-8 Group reports class diagram.....	IV-26
Figure IV-9 Assigned groups reports class diagram	IV-26
Figure IV-10 Statistics class diagram.....	IV-27
Figure IV-11 User permissions management class diagram	IV-28
Figure IV-12 Report submission class diagram	IV-29
Figure IV-13 Report modification class diagram.....	IV-30
Figure IV-14 Authentication sequence diagram	IV-31
Figure IV-15 Submission history sequence diagram	IV-32
Figure IV-16 Group reports sequence diagram.....	IV-32
Figure IV-17 Assigned groups reports sequence diagram	IV-33

Figure IV-18 Statistics sequence diagram.....	IV-33
Figure IV-19 User permission management sequence diagram.....	IV-34
Figure IV-20 Report submission sequence diagram	IV-34
Figure IV-21 Report modification sequence diagram.....	IV-35
Figure IV-22 ER diagram.....	IV-36
Figure IV-23 Firebase AppTester dashboard	IV-66
Figure IV-24 Crashlytics dashboard	IV-66
Figure V-1 Use case unit tests results	V-2
Figure V-2 Instrumental end-to-end tests results	V-4

Chapter I

INTRODUCTION

I.1 Introduction

STAIIS (STAI Imam Syafi'i) is an educational institution where the students are required to attend daily meetings. The management has been using manual methods to keep track of the students that violate these rules. The manual methods consume much more time and energy than needs to be when using a software to automate what can be automated. An application to automate the processing is developed with the principles and practices to keep it maintainable and scalable to continue to be used and to continue adapting with ease to new requirements as they are needed.

Software being easy to change compared to hardware is one of its defining characteristics. Software that cannot be easily changed cannot be adopted to new requirements as it grows in size and complexity. Maintaining software maintainability and ease of change are two of the responsibilities of software engineers as without these two, as software grows in size and complexity, if it the cost of maintaining it outweighs the profits, that would lead to the abandonment of projects and to the waste of the energy invested in developing them as doing otherwise is economically unviable for the stakeholders (Martin 2017).

Various attempts have been made, with varying degrees of success, to make the complexity of software manageable as it grows in size. Those attempts include the introduction of principles and programming paradigms that have been found by others to help them maintain the complexity of software manageable as it grows in size. The principles and paradigms to be demonstrated an application of in this thesis include a subset of the clean architecture principles, the MVVM (Model-View-ViewModel) architecture pattern, the declarative UI (User interface) programming paradigm, and the reactive programming

paradigm. All of these principles will be demonstrated an application of in this thesis with the aim of maintain the complexity of the software manageable as it grows in size and the aim of making it easy to change and adapt to new business requirements changes.

As a use case, an Android application will be developed for an educational institution in Jawa Barat, Indonesia which will help them with managing their students' data. In the piece of software to be developed, an application of the principles and paradigms mentioned above will be demonstrated.

I.2 Research Problem

Given the aim of this thesis, the research questions can be formulated as follows:

1. How are Clean and MVVM architecture patterns implemented in a Modern Android application?
2. How are Reactive programming and Declarative UI paradigms implemented in a Modern Android application?

I.3 Research Aims

1. The production of a mobile application that will be used on a daily basis by a group of supervisors at STAIIS.
2. Contributing to the literature by providing a demonstration of how Clean Architecture principles, MVVM architecture pattern, Reactive Programming and Declarative UI paradigms are implemented in a Modern Android application

I.4 Research Benefits

1. By the end of this project, the following is hoped to be achieved
2. Making the management of the students' data at STAIIS less time consuming
3. Producing a scalable and maintainable application

4. Making the implementation of the principles and paradigms mentioned above easier for other developers by providing an example of how that is done

I.5 Project Scope

The project scope will be limited to the following:

1. The application to be developed will be a native Android one
2. Testing will be limited to unit and instrumental tests that will be based on the use cases of the functional requirements

I.6 Writing Organizational System

The organizational system of writing of this thesis will follow the standard of the faculty of computer science of Sriwijaya University

I.6.1 Chapter 1

This chapter gives a background to the research and includes the problem statement, research aims, research benefits, and project scope.

I.6.2 Chapter 2

This chapter explains the concepts used in the following chapters

I.6.3 Chapter 3

This chapter explains the steps to be taken starting with the analysis all the way to the implementation

I.6.4 Chapter 4

This chapter includes steps taken to implement the application and the problems faced during the implementation of the application.

I.6.5 Chapter 5

This chapter includes the analysis of the results of the implementation and the analysis of the test results

I.6.6 Chapter 6

This chapter includes a summary of the research and suggestions for those who continue to take this further.

I.7 Summary

Clean Architecture principles, the MVVM paradigm, Reactive programming, and Declarative UI help developers create testable, maintainable, and scalable applications. As a demonstration of how those principles and paradigms come together to produce such applications, a native Android application for solving the problems the supervisors at STAIIS are facing with managing a subset of their students' data will be developed where all those principles and paradigms will be applied to develop an app that can continue to be developed and benefitted from by the supervisors at STAIIS.

DAFTAR PUSTAKA

- Android. 2023a. “LiveData Overview.” Retrieved February 21, 2023 (<https://developer.android.com/topic/libraries/architecture/livedata>).
- Android. 2023b. “Modern Android Development.” Retrieved February 2, 2023 (<https://developer.android.com/modern-android-development>).
- Dumbravan, Alexandru. 2022. *Clean Android Architecture*. Packt Publishing.
- Emets, V. F., and Jan Rogowski. 2018. “Comparative Analysis of Event Propagation Methods on Android Platform.” Pp. 227–30 in *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*. Vol. 1.
- Ghita, Catalin. 2022. *Kickstart Modern Android Development with Jetpack and Kotlin: Enhance Your Applications by Integrating Jetpack and Applying Modern App Architectural Concepts*.
- Künneth, Thomas. 2022. *Android UI Development with Jetpack Compose*. Packt Publishing.
- Len Bass, Paul Clements, Rick Kazman. 2021. *Software Architecture in Practice*. Pearson Education, Inc.
- Martin, Robert C. 2017. *Clean Architecture - A Craftsman’s Guide to Software Structure and Design*.
- Martin, Robert C. 2019. *Clean Agile - Back to Basics*. Pearson.
- Prem Ponuthorai, Jon Loeliger. 2022. *Version Control with Git - Powerful Tools and Techniques for Collaborative Software Development*. O’Reilly Media.
- Sandstø, Rebecca, and Cornelia Reme-Ness. 2021. “Agile Practices and Impacts on Project Success.” *Journal of Engineering, Project, and Production Management* 11(3):255–62. doi: 10.2478/jeppm-2021-0024.
- Sheikh, Waseem, and Nadeem Sheikh. 2020. “A Model-View-ViewModel (MVVM)

Application Framework for Hearing Impairment Diagnosis - Type Dependency Architecture.” Pp. 1–6 in *2020 Intermountain Engineering, Technology and Computing (IETC)*.

Statcounter. 2023. “Mobile Operating System Market Share Worldwide | Statcounter Global Stats.” Retrieved February 12, 2022 (<https://gs.statcounter.com/os-market-share/mobile/worldwide>).