

**TUGAS AKHIR**

**RANCANG BANGUN PLUGIN CLEAN ARCHITECTURE PROJECT  
TEMPLATE UNTUK ANDROID STUDIO MENGGUNAKAN  
METODE EXTREME PROGRAMMING**



**DISUSUN OLEH:**

**NANDA ARYA PUTRA**

**NIM. 21106050048**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA  
YOGYAKARTA**

**2025**

# LEMBAR PENGESAHAN



KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA  
FAKULTAS SAINS DAN TEKNOLOGI  
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

## PENGESAHAN TUGAS AKHIR

Nomor : B-1048/Un.02/DST/PP.00.9/06/2025

Tugas Akhir dengan judul : RANCANG BANGUN PLUGIN CLEAN ARCHITECTURE PROJECT TEMPLATE  
UNTUK ANDROID STUDIO MENGGUNAKAN METODE EXTREME  
PROGRAMMING

yang dipersiapkan dan disusun oleh:

Nama : NANDA ARYA PUTRA  
Nomor Induk Mahasiswa : 21106050048  
Telah diujikan pada : Senin, 26 Mei 2025  
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

## TIM UJIAN TUGAS AKHIR



Ketua Sidang

Dr. Agung Fatwanto, S.Si., M.Kom.  
SIGNED

Valid ID: 684a18325bfad



Penguji I

Nurochman, S.Kom., M.Kom.  
SIGNED

Valid ID: 68439701ef15a



Penguji II

Muhammad Galih Wonoseto, M.T.  
SIGNED

Valid ID: 684927e361d3f



Yogyakarta, 26 Mei 2025

UIN Sunan Kalijaga  
Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.  
SIGNED

Valid ID: 684a4eaddc66ed

# LEMBAR PERNYATAAN

## SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Nanda Arya Putra  
NIM : 21106050048  
Program Studi : Informatika

Menyatakan bahwa tugas akhir saya yang berjudul “Rancang Bangun *Plugin Clean Architecture Project Template* Untuk Android Studio Menggunakan Metode *Extreme Programming*” merupakan hasil penelitian saya sendiri, tidak terdapat karya di suatu perguruan tinggi, dan sepanjang sepengetahuan saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 5 Mei 2025



Nanda Arya Putra  
NIM. 21106050048

# LEMBAR PERSETUJUAN



Universitas Islam Negeri Sunan Kalijaga



FM-UINSK-BM-05-03/R0

## SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir

Lamp : -

Kepada  
Yth. Dekan Fakultas Sains dan Teknologi  
UIN Sunan Kalijaga Yogyakarta  
Di Yogyakarta

*Assalammu'alaikum wr. wb.*

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka saya selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Nanda Arya Putra  
NIM : 21106050048  
Judul Skripsi : Rancang Bangun Plugin Clean Architecture Project Template  
Untuk Android Studio Menggunakan Metode Extreme  
Programming

sudah dapat diajukan kembali kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara dapat segera dimunaqasyahkan. Atas perhatiannya saya ucapkan terima kasih.

*Wassalammu'alaikum wr. wb.*

Yogyakarta, 05 Mei 2025

Pembimbing

Dr. Agung Fatwanto, S.Si., M.Kom.

NIP. 19770103 200501 1 003

## **LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR**

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN Sunan Kalijaga Yogyakarta, diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen Tugas Akhir ini merupakan hak milik UIN Sunan Kalijaga Yogyakarta.

## INTISARI

Pengembangan aplikasi Android yang efisien memerlukan penerapan arsitektur perangkat lunak yang baik. Salah satu arsitektur yang banyak digunakan adalah *clean architecture*. Namun, implementasi *clean architecture* secara manual seringkali memakan waktu dan berisiko tinggi terhadap kesalahan manusia. Penelitian ini bertujuan untuk merancang dan membangun *plugin* Android Studio yang dapat menghasilkan template proyek berbasis *clean architecture* secara otomatis. Metode yang digunakan dalam pengembangan *plugin* ini adalah *Extreme Programming (XP)*. *Plugin* ini secara otomatis membuat struktur proyek, kode boilerplate, dependensi, dan konfigurasi sesuai dengan rekomendasi resmi Android. Hasil pengujian melalui *unit testing*, *black box testing*, dan *System Usability Scale (SUS)* menunjukkan bahwa *plugin* ini berfungsi dengan baik dan mendapatkan kepuasan pengguna yang cukup tinggi. *Plugin* ini dapat mempercepat pengembangan aplikasi Android dengan *clean architecture*, mengurangi risiko kesalahan, dan meningkatkan produktivitas pengembang.

**Kata Kunci:** *Plugin, Clean Architecture, Extreme Programming, Pengembangan Perangkat Lunak, System Usability Scale*

## ABSTRACT

Efficient Android application development requires the implementation of a good software architecture. One of the most commonly used architectures is clean architecture. However, manually implementing clean architecture often takes time and is prone to human errors. This research aims to design and build an Android Studio plugin that can automatically generate a project template based on clean architecture. The method used in the development of this plugin is Extreme Programming (XP). This plugin automatically creates project structure, boilerplate code, dependencies, and configurations in accordance with Android's official recommendations. Testing results through unit testing, black box testing, and System Usability Scale (SUS) show that this plugin works well and achieves high user satisfaction. This plugin can accelerate Android application development with clean architecture, reduce the risk of errors, and improve developer productivity.

**Keywords:** Plugin, Clean Architecture, Extreme Programming, Software Development, System Usability Scale

## **MOTTO**

"No tree, it is said, can grow to heaven unless its roots reach down to hell."

Carl Jung

“Sesungguhnya beserta kesulitan ada kemudahan.”

Surah Al-Insyirah ayat 6

"Rivers know this: There is no hurry. We shall get there someday."

Alan Alexander Milne

"A bird sitting on a branch is never afraid of the branch breaking, because its trust is not in the branch, but in its own wings."

Unknown

“Hidup memang tidak adil, Jadi biasakan dirimu.”

Patrick Star

## **HALAMAN PERSEMBAHAN**

Dengan penuh rasa syukur dan terima kasih kepada Allah yang Maha Pengasih dan Maha Penyayang, saya persembahkan Tugas Akhir ini sebagai tanda penghormatan dan rasa cinta yang mendalam. Semoga karya ini dapat bermanfaat dan menjadi amal yang baik. Tugas Akhir ini saya persembahkan dengan penuh cinta dan hormat kepada:

1. Ibu tercinta, Sulasmi, yang telah memberikan kasih sayang, doa, dan dukungan yang tiada henti. Engkau adalah sumber inspirasi dalam setiap langkah hidupku.
2. Almarhum Bapak tercinta, Sunarjan, yang telah menjadi pilar penopang dalam hidupku. Dengan kebijaksanaan dan kesabaranmu, Engkau membimbingku melewati segala rintangan dan meraih keberhasilan. Doa dan pengorbananmu adalah cahaya yang terus menyinari perjalanan hidupku.
3. Kedua adikku, Fassa Raditiya Putra dan Faith Chairil Putra, yang selalu menjadi sumber semangat dan motivasi bagi saya dalam menyelesaikan tugas akhir ini.
4. Ramjani Yaumi Safitri, yang setia mendampingi, memberikan dukungan, dan selalu menjadi pendengar yang baik dalam setiap proses penyusunan tugas akhir ini.
5. Semua sahabat dan teman seperjuangan yang telah memberikan dukungan dan semangat tanpa henti.

Terima kasih yang tak terhingga atas segala dukungan yang telah kalian berikan. Semoga Allah senantiasa melimpahkan rahmat dan berkah-Nya, serta memberikan balasan terbaik untuk semua kebaikan yang telah diberikan.

## KATA PENGANTAR

*Alhamdulillah Rabbil 'Alamin*, penulis mengucapkan syukur kepada Allah *Subhanahu wata'ala* atas rahmat dan karunia-Nya, yang memungkinkan penulis untuk menyelesaikan tugas akhir ini yang berjudul "*Rancang Bangun Plugin Clean Architecture Project Template Untuk Android Studio Menggunakan Metode Extreme Programming.*" Sholawat dan salam senantiasa tercurah kepada junjungan kita, Rasulullah SAW, yang telah membawa umat manusia ke dalam zaman yang penuh cahaya, dan semoga kita semua diberi syafaatnya di akhir zaman.

Penyelesaian tugas akhir ini memerlukan usaha dan kerja keras dalam setiap tahapannya. Namun, proses penyelesaian ini tidak terlepas dari bantuan dan dukungan berbagai pihak, baik yang terlibat langsung maupun tidak langsung. Pada kesempatan ini, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Prof. Noorhaidi Hasan S.Ag., M.A., M.Phil., Ph.D. selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Dr. Dra. Hj. Khurul Wardati, M.Si. selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Dr. Muhammad Mustakim, S.T. M.T. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Nurochman, S.Kom., M.Kom. selaku dosen penasihat akademik penulis selama kuliah di UIN Sunan Kalijaga.
5. Bapak Dr. Agung Fatwanto, S.Si., M.Kom. yang telah memberikan masukan berharga dan dukungan penuh dalam penyusunan tugas akhir ini, serta selalu sabar mengarahkan penulis dalam setiap tahapan.
6. Seluruh Dosen dan Karyawan Program Studi Informatika UIN Sunan Kalijaga yang telah memberi ilmu dan bantuan selama perkuliahan penulis.
7. Kedua orang tua penulis, Alm. Bapak Sunarjan dan Ibu Sulasmi yang dengan kasih sayang, doa, dan pengorbanan telah membesarkan serta mendidik penulis hingga dapat menempuh pendidikan tinggi dan menjadi pribadi yang lebih baik.
8. Kedua adik penulis, Fassa Raditiya Putra dan Faith Chairil Putra yang selalu menjadi sumber semangat dan motivasi penulis dalam menyelesaikan tugas akhir.

9. Ramjani Yaumi Safitri, yang telah setia menemani, memberikan dukungan, dan selalu mendengarkan setiap keluhan penulis sepanjang penyusunan tugas akhir ini.
10. Seluruh Angkatan Informatika terutama teman-teman angkatan 2021 yang telah menjadi teman seperjuangan dan saling mendukung pada saat perkuliahan.

Tugas akhir ini tentu masih memiliki kekurangan baik dari segi kualitas maupun kuantitas materi. Oleh karena itu, penulis sangat mengharapkan kritik dan saran konstruktif yang dapat membantu memperbaiki dan menyempurnakan karya ini. Semoga tugas akhir ini dapat memberikan manfaat dan dapat dikembangkan lebih lanjut di masa depan.

Yogyakarta, 1 Mei

2025

Penulis

Nanda Arya Putra

NIM. 21106050048

## DAFTAR ISI

|                                                                                                  |      |
|--------------------------------------------------------------------------------------------------|------|
| LEMBAR PENGESAHAN .....                                                                          | i    |
| LEMBAR PERNYATAAN .....                                                                          | ii   |
| LEMBAR PERSETUJUAN .....                                                                         | iii  |
| LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR .....                                                      | iv   |
| INTISARI .....                                                                                   | v    |
| ABSTRACT .....                                                                                   | vi   |
| MOTTO .....                                                                                      | vii  |
| HALAMAN PERSEMBAHAN .....                                                                        | viii |
| KATA PENGANTAR .....                                                                             | ix   |
| DAFTAR ISI .....                                                                                 | xi   |
| DAFTAR TABEL .....                                                                               | xiii |
| DAFTAR GAMBAR .....                                                                              | xiv  |
| BAB I PENDAHULUAN .....                                                                          | 1    |
| 1.1 Latar Belakang .....                                                                         | 1    |
| 1.2 Rumusan Masalah .....                                                                        | 2    |
| 1.3 Tujuan .....                                                                                 | 3    |
| 1.4 Manfaat .....                                                                                | 3    |
| BAB II TINJAUAN PUSTAKA .....                                                                    | 4    |
| 2.1 Clean Architecture .....                                                                     | 4    |
| 2.2 Implementasi <i>clean architecture</i> dalam aplikasi Android .....                          | 7    |
| 2.3 Pemetaan Prinsip Clean Architecture terhadap Implementasi Android dan Kebutuhan Plugin ..... | 10   |
| 2.4 Plugin .....                                                                                 | 12   |
| 2.5 Analisis Plugin Serupa .....                                                                 | 12   |
| BAB III METODE PENGEMBANGAN SISTEM .....                                                         | 16   |
| 3.1 Lokasi dan Waktu .....                                                                       | 16   |
| 3.2 Alat dan Bahan .....                                                                         | 16   |
| 3.3 Metode Pengembangan .....                                                                    | 16   |
| 3.4 Metode Pengujian .....                                                                       | 18   |
| 3.5 Tahapan Iterasi .....                                                                        | 19   |
| BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM .....                                                 | 20   |

|                                                                      |    |
|----------------------------------------------------------------------|----|
| 4.1 Iterasi Pengembangan Pertama .....                               | 20 |
| 4.1.1 Perencanaan ( <i>Planning</i> ) Iterasi Pertama .....          | 20 |
| 4.1.2 Perancangan ( <i>Design</i> ) Iterasi Pertama .....            | 21 |
| 4.1.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Pertama .....      | 25 |
| 4.1.4 Pengujian ( <i>Testing</i> ) Iterasi Pertama .....             | 26 |
| 4.1.5 Evaluasi <i>Acceptance Test Criteria</i> Iterasi Pertama ..... | 27 |
| 4.2 Iterasi Pengembangan Kedua.....                                  | 27 |
| 4.2.1 Perencanaan ( <i>Planning</i> ) Iterasi Kedua.....             | 28 |
| 4.2.2 Perancangan ( <i>Design</i> ) Iterasi Kedua.....               | 29 |
| 4.2.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Kedua.....         | 33 |
| 4.2.4 Pengujian ( <i>Testing</i> ) Iterasi Kedua.....                | 35 |
| 4.2.5 Evaluasi <i>Acceptance Test Criteria</i> Iterasi Kedua.....    | 37 |
| 4.3 Iterasi Pengembangan Ketiga .....                                | 38 |
| 4.3.1 Perencanaan ( <i>Planning</i> ) Iterasi Ketiga.....            | 38 |
| 4.3.2 Perancangan ( <i>Design</i> ) Iterasi Ketiga.....              | 40 |
| 4.3.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Ketiga .....       | 42 |
| 4.3.4 Pengujian ( <i>Testing</i> ) Iterasi Ketiga.....               | 43 |
| 4.3.5 Evaluasi <i>Acceptance Test Criteria</i> Iterasi Ketiga .....  | 48 |
| 4.4 Panduan Pengguna .....                                           | 49 |
| 4.5 Evaluasi <i>Plugin</i> .....                                     | 51 |
| BAB V KESIMPULAN DAN SARAN .....                                     | 59 |
| 5.1 Kesimpulan .....                                                 | 59 |
| 5.2 Saran .....                                                      | 59 |
| DAFTAR PUSTAKA.....                                                  | 60 |
| LAMPIRAN .....                                                       | 63 |

## **DAFTAR TABEL**

Tabel 2.1 Pemetaan Lapisan Clean Architecture Umum dengan Implementasi pada Android

Tabel 2.2 Pemetaan Prinsip dan Lapisan Clean Architecture Terhadap Kebutuhan Plugin

Tabel 2.3 Hasil Eksplorasi Plugin Clean Architecture pada JetBrains Marketplace

Tabel 4.1 Acceptance Test Criteria Iterasi Pertama

Tabel 4.2 Hasil Pengujian Black Box Untuk Iterasi Pertama

Tabel 4.3 Acceptance Test Criteria Iterasi Kedua

Tabel 4.4 Hasil Pengujian Black Box Untuk Iterasi Kedua

Tabel 4.5 Acceptance Test Criteria Iterasi Ketiga

Tabel 4.6 Hasil Pengujian Black Box Untuk Iterasi Ketiga

Tabel 4.7 Daftar Pernyataan System Usability Scale (SUS)

Tabel 4.8 Data Awal Hasil Pengujian SUS

Tabel 4.9 Hasil Perhitungan Skor SUS

Tabel 4.10 Skala Penilaian SUS Jeff Sauro

## **DAFTAR GAMBAR**

- Gambar 2.1 Bentuk Umum Clean Architecture
- Gambar 3.1 Tahapan Metode Extreme Programming
- Gambar 4.1 Component Diagram Plugin
- Gambar 4.2 Use Case Diagram Plugin
- Gambar 4.3 Template Proyek Plugin Terlihat di New Project
- Gambar 4.4 Activity Diagram membuat proyek baru
- Gambar 4.5 Wireframe Tampilan Pertama Plugin
- Gambar 4.6 Wireframe Tampilan Kedua Plugin
- Gambar 4.7 Tampilan Antarmuka Pertama Template Proyek Plugin
- Gambar 4.8 Tampilan Antarmuka Kedua Template Proyek Plugin
- Gambar 4.9 Struktur Pohon Direktori Proyek Android Berbasis Clean Architecture
- Gambar 4.10 Proyek yang Dihasilkan oleh Template Plugin
- Gambar 4.11 Hasil Pengujian Fungsionalitas dengan Unit Testing
- Gambar 4.12 Hasil Code Coverage dari Pengujian Unit Testing
- Gambar 4.13 Plugin Berhasil Terpasang di Android Studio

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam dekade terakhir, manusia telah mengalami perkembangan teknologi yang semakin cepat. Hal ini termasuk peningkatan jumlah pengguna *smartphone* setiap tahunnya. Pengguna *smartphone* telah mencapai 5,78 miliar orang di dunia pada awal 2025. Jumlah ini setara dengan 70,5% dari total populasi dunia. Selain itu, pengguna *smartphone* telah meningkat sebesar 112 juta dalam 12 bulan terakhir [1]. Pertumbuhan pesat pengguna *smartphone* membuka peluang besar bagi pengembang aplikasi Android. Selain itu, meningkatnya jumlah pengguna *smartphone* turut meningkatkan kebutuhan aplikasi Android yang semakin kompleks. Hal ini mendorong para pengembang untuk menciptakan aplikasi yang lebih efisien, modular, dan mudah dikembangkan.

Dalam pengembangan aplikasi, pengembang cenderung menambah *class* baru daripada mengubah yang lama, sehingga menyebabkan ukuran aplikasi membesar seiring waktu [2]. Hal ini berdampak pada meningkatnya kompleksitas aplikasi, yang mempersulit proses pemeliharaan aplikasi. Dengan demikian, pemilihan arsitektur yang tepat menjadi krusial. Arsitektur yang baik mempermudah pemeliharaan, perbaikan bug, pengujian, optimasi aplikasi, serta mempercepat waktu rilis [3]. Tanpa arsitektur yang terstruktur, pengembang berisiko menghadapi peningkatan biaya pengembangan dan potensi kegagalan proyek akibat kompleksitas kode yang sulit dikelola.

*Clean architecture* adalah salah satu pendekatan arsitektur perangkat lunak yang menekankan pemisahan tanggung jawab dalam sistem ke dalam lapisan-lapisan dengan tugas spesifik. Pendekatan ini meningkatkan fleksibilitas, memudahkan pengujian, serta memungkinkan pengembang untuk mengganti komponen eksternal tanpa memerlukan banyak usaha [4]. Dengan karakteristik tersebut, *clean architecture* dapat membantu mengatasi tantangan kompleksitas dalam pengembangan aplikasi.

Sudah dibuktikan bahwa penggunaan *clean architecture* dalam pengembangan aplikasi android dapat memberikan tingkat *maintainability* 78.2% sampai 79% [5] [6]. Hal ini lebih baik daripada penggunaan arsitektur lain seperti MVC yang hanya memberikan tingkat *maintainability* 54,8%. Selain itu, arsitektur seperti MVC, MVP, MVVM, MVI, dan MVU hanyalah pola desain tingkat tinggi, bukan arsitektur global yang cukup untuk aplikasi mobile yang kompleks. Oleh karena itu, diperlukan arsitektur yang lebih menyeluruh, seperti

*clean architecture*, untuk membangun aplikasi yang lebih kompleks dan berkelanjutan [7].

Mengimplementasikan *clean architecture* dalam pengembangan aplikasi Android secara manual bukanlah tugas yang mudah. Pengembang harus secara mandiri menginisialisasi struktur proyek, membuat implementasi lapisan-lapisan seperti Domain, Data, dan Presentation, serta memastikan bahwa dependensi antar-lapisan tidak melanggar prinsip *clean architecture* [8]. Lapisan-lapisan tersebut sesuai dengan rekomendasi tim Android, sehingga memiliki validitas tinggi dalam praktik pengembangan aplikasi [9]. Namun, proses inisiasi *clean architecture* secara manual membutuhkan waktu yang signifikan dan berisiko tinggi terhadap kesalahan manusia, terutama dalam pengelolaan kode dan dependensi.

Untuk mengatasi tantangan tersebut, sebenarnya telah ada beberapa solusi dalam bentuk *plugin* yang dirancang untuk otomatisasi implementasi *clean architecture*. Namun, *plugin-plugin* yang sudah ada masih memiliki kekurangan dan belum cukup efisien. Beberapa kekurangannya diantaranya adalah bukan untuk platform android, hanya membuat direktori kosong tanpa *boiler code*, *template* proyek yang dibuat tidak sesuai rekomendasi resmi, dan kurangnya dependensi dalam *template* proyek yang dihasilkan.

Hal tersebutlah yang memotivasi penulis untuk merancang dan membangun *plugin* yang dapat membuat *clean architecture project template* untuk platform android sehingga struktur proyek, kode *boilerplate*, dependensi, dan konfigurasi proyek akan secara otomatis dibuat secara lengkap. *Plugin* ini dirancang untuk mengatasi kekurangan yang ada pada *plugin* serupa sebelumnya. Dengan demikian, waktu inisiasi proyek dengan *clean architecture* akan berkurang secara signifikan dan menghilangkan risiko kesalahan manusia dalam pengelolaan kode dan dependensi. *Plugin* ini akan dikembangkan dengan metode *extreme programming* dan dapat diintegrasikan ke dalam Android Studio, IDE resmi untuk pengembangan aplikasi Android.

## 1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan, dapat dirumuskan masalah sebagai berikut: “Bagaimana merancang dan membangun *plugin* Android Studio untuk mempercepat dan mempermudah pembuatan *template* proyek berbasis *clean architecture* dengan bahasa pemrograman Kotlin?”

### 1.3 Tujuan

Tugas akhir ini bertujuan untuk merancang dan membangun *plugin* Android Studio menggunakan metode Extreme Programming (XP) guna menghasilkan template proyek *clean architecture* dengan bahasa pemrograman Kotlin.

### 1.4 Manfaat

Tugas akhir ini diharapkan memberikan manfaat sebagai berikut:

- Meningkatkan produktivitas pengembang dalam membangun aplikasi Android.
- Mengurangi risiko kesalahan dalam implementasi *clean architecture*.
- Mempercepat proses pengembangan aplikasi dengan otomatisasi *template* proyek.
- Meningkatkan standar kualitas kode dalam pengembangan aplikasi Android.
- Mempermudah proses pemeliharaan kode dengan struktur yang lebih terorganisir.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penulis telah berhasil merancang dan membangun sebuah plugin untuk Android Studio yang dapat menghasilkan template proyek berbasis *clean architecture* secara otomatis. Dengan menggunakan metode *extreme programming* (XP), *plugin* ini dirancang untuk mempercepat dan mempermudah proses inisiasi proyek aplikasi android. Penggunaan *plugin* ini diharapkan dapat mengurangi risiko kesalahan manusia dalam mengelola kode dan dependensi, serta mempercepat proses pengembangan aplikasi android terutama implementasi *clean architecture*. *Plugin* ini mengotomatiskan pembuatan struktur proyek, kode boilerplate, dependensi, dan konfigurasi yang sesuai dengan dokumentasi resmi Android, yang sebelumnya membutuhkan upaya manual yang memakan waktu dan rawan kesalahan.

Hasil pengujian yang dilakukan melalui *unit testing*, *black box testing*, dan *System Usability Scale* (SUS) menunjukkan bahwa *plugin* ini berfungsi dengan baik dalam memenuhi tujuan yang telah ditetapkan. Dengan *code coverage* yang hampir mencapai 100% pada *class* dan *method*, serta hasil SUS yang menunjukkan tingkat kepuasan pengguna yang cukup tinggi, *plugin* ini dapat dianggap siap digunakan untuk mempercepat pengembangan proyek Android berbasis *clean architecture*. Secara keseluruhan, *plugin* ini berhasil mengatasi tantangan dalam implementasi *clean architecture* dan dapat meningkatkan produktivitas pengembang aplikasi Android.

#### 5.2 Saran

Beberapa saran yang dapat diberikan untuk pengembangan *plugin* lebih lanjut adalah sebagai berikut:

1. Menambahkan fitur tambahan lain seperti Firebase, SQLite, Datastore, dan fitur lainnya.
2. Menambahkan fitur untuk menghasilkan template proyek dengan antarmuka pengguna (UI) berbasis Jetpack Compose.
3. Menambahkan pilihan template proyek dengan bahasa pemrograman Java.
4. Mengoptimalkan kode sumber *plugin* agar lebih efisien dan mudah dipelihara.
5. Memperbaiki UI *plugin* agar lebih mudah digunakan dan lebih responsif.

## DAFTAR PUSTAKA

- [1] S. Kemp, “Digital 2025: Global Overview Report,” *datareportal.com*, 2025. <https://datareportal.com/reports/digital-2025-global-overview-report> (accessed Feb. 21, 2025).
- [2] J. Gao, L. Li, T. F. Bissyande, and J. Klein, “On the evolution of mobile app complexity,” *Proc. IEEE Int. Conf. Eng. Complex Comput. Syst. ICECCS*, vol. 2019-Novem, pp. 200–209, 2019, doi: 10.1109/ICECCS.2019.00029.
- [3] R. Verdecchia, I. Malavolta, and P. Lago, “Guidelines for architecting android apps: A mixed-method empirical study,” *Proc. - 2019 IEEE Int. Conf. Softw. Archit. ICSA 2019*, pp. 141–150, 2019, doi: 10.1109/ICSA.2019.00023.
- [4] R. C. Martin, *Clean architecture: a craftsman’s guide to software structure and design*. Prentice Hall Press, 2017.
- [5] M. H. Badrudduja and R. E. Putra, “Penerapan Clean Architecture pada Aplikasi Pemesanan Makanan menggunakan Metode Slope One Algorithm,” *J. Informatics Comput. Sci.*, vol. 3, no. 04, pp. 506–514, 2022, doi: 10.26740/jinacs.v3n04.p506-514.
- [6] M. Z. Azziqra and I. Nuryasin, “Implementasi Clean Architecture Pada Aplikasi Mobile Al-Quran Berbasis Flutter,” *JOISIE (Journal Inf. Syst. Informatics Eng.)*, vol. 8, no. 2, pp. 369–380, 2024, doi: 10.35145/joisie.v8i2.4763.
- [7] R. Nunkesser, “Choosing a Global Architecture for Mobile Applications.” TechRxiv, pp. 1–5, 2021. doi: 10.36227/techrxiv.14212571.v1.
- [8] A. S. Dewari, “Implementing Clean Architecture in an Android app,” *Medium*, 2024. <https://medium.com/@ajay.dewari/implementing-clean-architecture-in-an-android-app-20ff2af8b418> (accessed Feb. 23, 2025).
- [9] Android Developers, “Panduan arsitektur aplikasi,” *developer.android.com*, 2025. <https://developer.android.com/topic/architecture?hl=id> (accessed Feb. 23, 2025).
- [10] A. T. Sondha, U. Sa’adah, F. F. Hardiansyah, and M. B. A. Rasyid, “Framework dan Code Generator Pengembangan Aplikasi Android dengan Menerapkan Prinsip Clean Architecture,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 4, pp. 327–335, 2020, doi: 10.22146/jnteti.v9i4.572.
- [11] Android Developers, “Lapisan Data,” *developer.android.com*, 2024. <https://developer.android.com/topic/architecture/data-layer?authuser=1&hl=id>

(accessed Mar. 05, 2025).

- [12] Android Developers, “Lapisan Domain,” *developer.android.com*, 2024. <https://developer.android.com/topic/architecture/domain-layer?hl=id> (accessed Mar. 05, 2025).
- [13] H. Fakher, “Clean Architecture Applied to Android Development,” *Medium*, 2024. <https://medium.com/@husayn.fakher/clean-architecture-applied-to-android-development-de9132b5faf2> (accessed Mar. 05, 2025).
- [14] Android Developers, “Lapisan UI,” *developer.android.com*, 2024. <https://developer.android.com/topic/architecture/ui-layer?hl=id> (accessed Mar. 05, 2025).
- [15] JetBrains, “Android Studio Plugin Development,” *plugins.jetbrains.com*, 2024. <https://plugins.jetbrains.com/docs/intellij/android-studio.html> (accessed Mar. 05, 2025).
- [16] R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan,” *J. Sist. dan Teknol. Inf.*, vol. 8, no. 3, pp. 272–277, 2020, doi: 10.26418/justin.v8i3.40273.
- [17] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, “A Systematic Review on Extreme Programming,” *J. Phys. Conf. Ser.*, vol. 1969, no. 1, 2021, doi: 10.1088/1742-6596/1969/1/012046.
- [18] I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, “Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android,” *INOVTEK Polbeng - Seri Inform.*, vol. 5, no. 2, pp. 297–307, 2020, doi: 10.35314/isi.v5i2.1654.
- [19] Divyani Shivkumar Taley, “Comprehensive Study of Software Testing Techniques and Strategies: A Review,” *Int. J. Eng. Res.*, vol. V9, no. 08, pp. 817–822, 2020, doi: 10.17577/ijertv9is080373.
- [20] Supriyono, “Software Testing with the approach of Blackbox Testing on the Academic Information System,” *Int. J. Inf. Syst. Technol.*, vol. 3, no. 36, pp. 227–235, 2020, doi: 10.30645/IJISTECH.V3I2.54.
- [21] A. Bangor, P. Kortum, and J. Miller, “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale,” *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009, [Online]. Available: <https://dl.acm.org/doi/10.5555/2835587.2835589>

- [22] K. Hamilton and R. Miles, *Learning UML 2.0*. Sebastopol, California: O'Reilly Media, Inc., 2006.
- [23] A. Almani and O. Alrwais, "The Role of Wireframes in Enhancing User Interface Design," *Int. Res. J. Innov. Eng. Technol.*, vol. 8, no. 12, pp. 134–140, 2024.
- [24] A. M. Adam, "Sample Size Determination in Survey Research," *J. Sci. Res. Reports*, vol. 26, no. 5, pp. 90–97, 2020, doi: 10.9734/jsrr/2020/v26i530263.
- [25] JetBrains, "Data Playground," *jetbrains.com*, 2025. [https://www.jetbrains.com/lp/devecosystem-data-playground/#global\\_population](https://www.jetbrains.com/lp/devecosystem-data-playground/#global_population) (accessed Apr. 27, 2025).
- [26] A. Wulansari, A. Farqi, and T. L. M. Suryanto, "Analisis Tingkat Usabilitas Website Perguruan Tinggi Menggunakan System Usability Scale (SUS)," *J. Teknol. Inf. dan Terap.*, vol. 10, no. 2, pp. 126–131, 2023, doi: 10.25047/jtit.v10i2.328.
- [27] A.-M. Simundic, "Confidence interval," *Biochem. Medica*, vol. 18, no. 2, pp. 154–161, 2008, doi: 10.11613/BM.2008.015.
- [28] J. Sauro, "5 Ways to Interpret a SUS Score," *measuringu.com*, 2018. <https://measuringu.com/interpret-sus-score/> (accessed Apr. 29, 2025).