

ABSTRACT

The Colega application is a mobile-based prototype of a pedagogical agent designed to motivate Telkom University students by reminding them of academic activities from Moodle Learning/LMS, such as Assignments, Quizzes, and Class Schedules. However, the application has limitations in several aspects, such as restricted reminder settings and suboptimal navigation, which prevent users from maximizing its functionality as an effective academic assistance tool.

Colega is developed using Android Native with Kotlin and implements Clean Architecture to improve code quality. This architectural approach divides the code into three layers: Presentation, Domain, and Data, enhancing modularity, scalability, and maintainability. This structured approach facilitates testing, maintenance, and future feature development.

After development, the application underwent Unit Testing using JUnit and UI Testing using Espresso, with test scenarios covering Notification Features, Login and Logout, and Task Management. Additionally, SonarQube was used to evaluate code maintainability, with the results indicating that the Quality Gate Status: Passed, meaning the code meets the required quality standards. Despite 83 open issues, the application still received an A rating, signifying that the code remains easily maintainable.

By implementing Clean Architecture, validating with SonarQube, and conducting automated testing with JUnit and Espresso, this research aims to evaluate how a well-structured architecture can enhance code maintainability and quality in mobile application development.

Keywords: *E-learning, Activities, Learning Management System, Android, Activity Reminders, Clean Architecture, SonarQube*