

# Efek Design Pattern Terhadap *Duplicated Code* dan Efek Terusannya Pada *Maintainability* Aplikasi Berbasis *Mobile*

Firdaus Ardhana Indradhirmaya<sup>1</sup>, Dawam Dwi Jatmiko Suwawi<sup>2</sup>, Shinta Yulia Puspitasari<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>firdausai@students.telkomuniversity.ac.id, <sup>2</sup>dawamdjs@telkomuniversity.ac.id,

<sup>3</sup>shintayulia@telkomuniversity.ac.id

---

## Abstrak

Kemajuan teknologi mobile yang sangat pesat tidak lepas dari berbagai macam permasalahan, khususnya pengabaian proses pengembangan aplikasi mobile yang ideal. Hal tersebut telah menimbulkan berbagai macam masalah, salah satunya adalah meningkatnya jumlah *duplicate code*, sebuah permasalahan yang sering terjadi pada aplikasi *mobile* berbasis android. Selibuhnya, hal tersebut juga mengakibatkan menurunnya tingkat *maintainability* pada sebuah aplikasi. Penelitian-penelitian sebelumnya menyatakan bahwa pembuatan *abstract class* dapat mengatasi permasalahan *duplicate code*, namun juga menurunkan tingkat *maintainability*. Tujuan penelitian ini adalah untuk menerapkan dan mengamati efek *design pattern*, yang melibatkan pembuatan *abstract class*, untuk mengatasi permasalahan *duplicate code* dan juga melihat efek terusannya pada *maintainability* sebuah aplikasi. Metodologi penelitian yang telah dilakukan adalah membandingkan tingkat *maintainability* dan jumlah *duplicate code* sebelum dan sesudah penerapan *design pattern* terpilih. Jumlah baris *duplicate code* dan ISO 25010 akan digunakan sebagai metrik *duplicate code* dan *maintainability* masing-masing. Penerapan *template pattern* terbukti mampu untuk menekan jumlah *duplicate code*, namun gagal untuk meningkatkan atau mempertahankan tingkat *maintainability* secara keseluruhan. Fitur *inheritance* yang digunakan oleh *template pattern* akan selalu mengakibatkan metrik *depth of inheritance* dan *coupling* untuk memburuk. Selibuhnya, efek dari kode yang di abstraksi memungkinkan terjadinya pemburukan pada metrik *cohesion*, *complexity*, dan *number of methods*. Dimana metrik-metrik yang disebutkan berpengaruh terhadap tingkat *maintainability*, yang meliputi aspek *reusability*, *modifiability*, *modularity*, *testability* dan *analysability*.

Kata kunci : *design pattern*, *duplicate code*, *maintainability*, *mobile*

---

## Abstract

The rapid advancement of mobile technology brings a variety of issues along with it. One of those issues is neglecting the ideal mobile application development process. Such neglect has caused the increased number of duplicate codes, the most occurring issue in the android application. Furthermore, it also has caused the level of maintainability in mobile apps to drop. Previous studies have shown that abstract classes can decrease code duplication while also decrease the level of maintainability. This research aims to see the effects of implementing design patterns, most of which involve creating abstract class, to reduce duplicate code while also observing its subsequent effects on the maintainability aspect of a mobile application. The amount of duplicate code and the maintainability aspect were measured before and after the implementation of design pattern. In addition, the number of lines of duplicate code and ISO 25010 were referenced and used as metrics to measure duplicate code and maintainability, respectively. The template pattern was proven to reduce duplicate codes but could not maintain or increase the maintainability aspect of an android mobile application as a whole. The depth of inheritance and coupling metric will always worsen due to the use of inheritance by the template pattern. Furthermore, the effect of the code being abstracted could negatively affect the cohesion, complexity, and number of methods metric. The metrics mentioned contributed to the reusability, modifiability, modularity, testability, and analysability aspect of maintainability.

Keywords: *design pattern*, *duplicate code*, *maintainability*, *mobile*

---

## 1. Introduction

### Background

The rapid growth of mobile technology has been inseparable from various challenges in the last ten years [1]. The increasing demand and complex features in mobile applications cause developers to put aside quality