

Abstract

Design Pattern is a solution to solving problems in object-oriented programming. Problem solving done by the learning of a similar problem had happened before. So the solution will be formulated to solve the problems and solutions will be used to handle similar problems that occur in the future.

There are various kinds of design pattern, one of which is the facade pattern. Facade pattern is one of the structural design pattern, which is the part that discusses the relationship between classes or objects. Facade is method to simplifying the process by calling the client class (Graphical User Interface class) to logic class.

The title of this final project is *Improving Object-Oriented Software Design With the Facade Pattern*. The software will be used as the material is the library system. The analysis will be done by comparing the quality of the program that was built by applying the facade pattern along with a program designed only with the principles of object-oriented.

At the beginning, this project start with designing the software system requirements, then that system requirement is modeled using UML (Unified Modeling Language). Having completed the design process, will be implementing into the programming language. After the implementation, that software be analyzed using Object-Oriented Metrics to know the quality of the performance of the software.

Results showed that application of facade pattern, the software that implements facade will be more flexible to be developed. It is caused by low levels of coupling between subsystems classes with classes on software clients with facade pattern.

Keywords: *Design Pattern, Facade Pattern, Object, Complexity*