## Abstract

Inheritance is an object-oriented concept that supports reusability. The attributes or methods of existing class can be extended or used by classes that have similarity without having to redefine. However, when this concept is applied in case that requires additional responsibility of an object dynamically, it will cause other problems happen such as class explosion, rigid design, or adding functionality to base class that isn't appropriate for some of the subclasses. Decorator pattern provides alternative to subclassing for extending functionality of an object dynamically at program runtime.

This final project analyzes the implementation of class design that uses decorator pattern in a *Grid-Paging* application on *ExtJS*. Then the design implementation is measured its complexity, productivity, and reusability using object-oriented metrics and so does its execution time. After that, the measurement results are compared with the measurement of an application that is not designed using decorator pattern which has the same functionality to know the advantages and disadvantages of the implementation of decorator pattern.

After analyzing the metrics of both applications, it can be concluded that the application that uses decorator pattern had better reusability, so its complexity become higher, but its productivity become more expensive and also having longer execution time compared with application that does not use the decorator pattern.

Keywords: inheritance, decorator pattern, Grid-Paging, ExtJS