

## Abstrak

*Design Pattern* adalah sebuah solusi pemecahan masalah pada pemrograman berorientasi objek. Pemecahan masalah dilakukan dengan melakukan pembelajaran terhadap masalah serupa yang telah terjadi sebelumnya. Jadi akan dirumuskan solusi untuk menyelesaikan masalah dan solusi tersebut akan digunakan untuk menangani masalah serupa yang terjadi pada waktu mendatang.

Terdapat berbagai macam *design pattern*, salah satunya adalah *facade pattern*. *Facade pattern* merupakan salah satu *design pattern* struktural, yaitu bagian yang membahas hubungan atau relasi antar kelas atau objek. *Facade* adalah salah satu cara menyederhanakan proses pemanggilan yang dilakukan kelas klien (kelas *Graphical User Interface*) terhadap kelas logik.

Tugas akhir ini berjudul *Perbaikan Desain Perangkat Lunak Berorientasi Objek Dengan Facade Pattern*. Perangkat lunak yang akan dijadikan sebagai bahan analisis adalah sistem perpustakaan. Analisis akan dilakukan dengan membandingkan kualitas program yang dibangun dengan menerapkan *facade pattern* dengan program yang dirancang hanya dengan prinsip berorientasi objek saja.

Di awali dengan melakukan perancangan kebutuhan sistem perangkat lunak, kemudian kebutuhan tersebut dimodelkan menggunakan UML (*Unified Modelling Language*). Setelah poses perancangan selesai, akan dilakukan pengimplementasian ke dalam bahasa pemrograman. Setelah dilakukan implementasi kemudian akan dilakukan dianalisis menggunakan *Object Oriented Metrics* untuk mengetahui kualitas dari kinerja perangkat lunak tersebut.

Hasil penerapan *facade pattern* menunjukkan bahwa, perangkat lunak yang menerapkan facade akan lebih fleksibel untuk dikembangkan. Hal tersebut disebabkan karena rendahnya tingkat *coupling* antara kelas-kelas subsistem dengan kelas klien pada perangkat lunak dengan *facade pattern*.

**Kata kunci:** *Design Pattern, Facade Pattern, Objek, Kerumitan*