

Abstrak

Design pattern muncul akibat adanya permasalahan yang sama yang sering muncul pada desain pembuatan perangkat lunak. Pada perkembangannya sudah banyak *design pattern* yang sudah ditemukan oleh para programmer. Saat ini, *design pattern* dikelompokkan ke dalam tiga tujuan berbeda, yaitu *creational*, *structural*, dan *behavioral*.

Observer pattern merupakan salah satu *design pattern* yang tergolong ke dalam *behavioral pattern*. *Observer pattern* muncul akibat adanya kondisi di mana satu state akan dikirimkan ke beberapa objek, atau bahasa lainnya adalah *broadcast communication*. *Pattern* ini memungkinkan satu state dikirim ke berbagai objek yang terdependensi dengan coupling yang rendah antara objek satu dengan objek lainnya.

Pada tugas akhir kali ini, dibuat sebuah perangkat lunak yang mengimplementasikan *observer pattern* untuk menyelesaikan sebuah kasus yang memerlukan *broadcast communication*. Untuk mengevaluasi *observer pattern*, dilakukan pengujian serta perhitungan terhadap *object-oriented metrics*. Sedangkan untuk mengetahui kelebihan dan kekurangan yang ada pada *observer pattern*, akan dibandingkan hasil perhitungan *object-oriented metrics* perangkat lunak yang menerapkan *observer pattern* dengan perangkat lunak yang tidak menerapkan *observer pattern* untuk sebuah kasus yang serupa.

Kata kunci: design pattern, observer pattern, dan object-oriented metrics.