

Abstrak

Iterator pattern merupakan salah satu *design pattern* yang tergolong ke dalam *behavioral pattern*. *Iterator pattern* ini digunakan dalam mengakses sekumpulan objek tanpa menyingkap struktur di dalamnya. *Pattern* ini mengambil alih mekanisme iterasi ke dalam kelas *iterator*. Pada tugas akhir ini, dibuat sebuah perangkat lunak yang mengimplementasikan *iterator pattern*. Untuk mengukur kualitas *iterator pattern*, dilakukan pengujian serta perhitungan dengan *object-oriented metrics* dan *traditional metrics*, yaitu WMC, CBO, LCOM, DIT, EXEC, dan MHF. Lalu untuk mengetahui kelebihan dan kekurangan pada *iterator pattern*, akan dibandingkan hasil perhitungan *metrics* perangkat lunak yang menerapkan *iterator pattern* dengan perangkat lunak yang tidak menerapkan *iterator pattern* untuk kasus yang sama, untuk tugas akhir ini akan digunakan studi kasus *Newsreader*.

Berdasarkan hasil analisis dan pengujian, jika dilihat dari perhitungan *metrics*, maka perangkat lunak dengan *iterator pattern* memiliki kompleksitas yang lebih tinggi sehingga dibutuhkan usaha lebih besar ketika dilakukan *maintenance* dibandingkan perangkat lunak tanpa *iterator pattern*. Namun ketika terjadi penambahan kelas yang menciptakan objek container, untuk menerapkan mekanisme iterasi pada perangkat lunak yang menerapkan *iterator pattern* lebih mudah dilakukan dan dengan menerapkan *iterator pattern* akan menyelesaikan masalah enkapsulasi dan efisiensi penulisan kode pada perangkat lunak yang mengimplementasikan mekanisme iterasi di dalamnya.

Kata kunci: *design pattern*, *iterator pattern*, dan *metrics*.