

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*.

Factory Method Pattern merupakan salah satu *design pattern* yang termasuk dalam kelompok *creational pattern*. *Creational pattern* adalah kelompok *design pattern* yang berhubungan dengan pembuatan objek. *Factory method pattern* digunakan untuk memisahkan (*decouple*) proses pembuatan/instansiasi sebuah objek (produk) dari objek lain (klien) yang menggunakannya. Tujuannya supaya perubahan pada *product class* tidak menyebabkan perubahan kode pada *client*. Paling tidak akibat dari perubahan itu bisa diminimalisir. Dan juga supaya *factory* bisa digunakan oleh banyak kelas.

Pada tugas akhir kali ini, dibuat sebuah perangkat lunak yang mengimplementasikan *factory method pattern* untuk menyelesaikan sebuah kasus. Untuk mengevaluasi *factory method pattern*, digunakan *metrics* pengukuran yang khusus menangani perangkat lunak berorientasi objek yaitu *object-oriented metrics*. Hasil pengujian ini kemudian dibandingkan dengan hasil pengujian perangkat lunak yang tidak menerapkan *factory method pattern*. Pada akhirnya didapatkan kesimpulan tentang kualitas kedua perangkat lunak dalam hal *reusability*.

Berdasarkan hasil pengujian didapatkan bahwa perangkat lunak dengan *factory method pattern* memiliki *reusability* yang lebih baik bila dibandingkan dengan perangkat lunak tanpa *factory method pattern*. Hal ini berarti pengembangan perangkat lunak yang menerapkan *factory method pattern* akan lebih mudah dibandingkan dengan perangkat lunak tanpa *factory method pattern*.

Kata kunci: *design pattern, factory method pattern, reusability, dan object-oriented metrics.*