

# 1. PENDAHULUAN

## 1.1 Latar belakang masalah

Pemrograman berorientasi objek (PBO) adalah salah satu cara membangun aplikasi, dengan didasarkan pada objek-objek dunia nyata, yang mengandung data dan fungsi/prosedur yang bekerja atas objek tersebut. Dalam perkembangannya, banyak masalah-masalah yang kerap terjadi dalam pembangunan aplikasi berorientasi objek. Kebanyakan masalah tersebut memiliki inti permasalahan yang sama, yang telah diselesaikan oleh para pembuat program [1].

*Design pattern* muncul akibat adanya permasalahan yang sama yang sering muncul pada desain pembuatan perangkat lunak [4]. Dengan menggunakan *design pattern*, tingkat *reusability* suatu *design* akan semakin tinggi. 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* [4].

*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. Pemisahan ini menjadikan desain perangkat lunak dengan *factory method pattern* lebih mudah dikembangkan dan digunakan kembali pada aplikasi lain [12]. Dengan kata lain, penerapan *factory method pattern* membuat desain lebih *reusable*.

Untuk mengukur tingkat *reusability* ini, digunakan *metric* pengukuran yang secara khusus menangani pengukuran perangkat lunak berorientasi objek. *Metric* ini dikenal dengan sebutan *object oriented metrics (OO Metrics)*. Atribut pengukuran yang dapat uji dengan *OO Metrics* ini antara lain: *complexity*, *reusability*, *coupling*, dan *cohesion*.

## 1.2 Perumusan masalah

Permasalahan yang akan dibahas pada tugas akhir ini adalah sebagai berikut:

1. Apakah penggunaan *factory method pattern* bisa meningkatkan *reusability* suatu perangkat lunak?
2. Bagaimana kualitas suatu perangkat lunak yang mengimplementasi *factory method pattern* dan perangkat lunak yang tidak mengimplementasikan *factory method pattern* dalam hal *reusability* berdasarkan *oo metrics*?

## 1.3 Batasan masalah

Batasan masalah pada tugas akhir ini adalah :

1. Komponen *OO metrics* yang di analisis :
  - a. Coupling Between Object Classes (CBO)
  - b. Lack of Cohesion of Methods (LCOM)
  - c. Weighted Methods per Class (WMC)

2. Menggunakan bahasa pemrograman Java dan NetBeans IDE Dev sebagai aplikasi GUI nya serta Microsoft Windows 7 sebagai system operasinya.
3. Desain perangkat lunak dimodelkan dengan menggunakan Microsoft Visio 2003.
4. Aplikasi yang dibuat berbasis desktop, studi kasus: sistem informasi kos-kosan.

## 1.4 Tujuan

Tujuan yang ingin dicapai dalam pembuatan Tugas Akhir ini antara lain :

- a) Menerapkan *factory method pattern* pada sebuah perangkat lunak berbasis desktop.
- b) Menganalisa dan membandingkan *reusability* dari kedua perangkat lunak yang menerapkan *factory method pattern* dan yang tidak menerapkan *factory method pattern* berdasarkan *OO metrics*.

## 1.5 Metodologi penyelesaian masalah

1. Studi Literatur :  
Mencari informasi dari berbagai sumber seperti: jurnal, buku, pencarian melalui internet, dan artikel-artikel mengenai beberapa topik seperti: *design pattern* dan *factory method pattern*.
2. Analisis dan Desain :  
Menganalisa dan merancang desain untuk membangun aplikasi desktop yang menerapkan *factory method pattern* dan yang tidak menerapkan *factory method pattern*.
3. Implementasi sistem:  
Membangun perangkat lunak yang menerapkan *factory method pattern* dan yang tidak menerapkan *factory method pattern*.
4. Analisis hasil implementasi:  
Analisis *OO metrics* terhadap aplikasi yang telah menerapkan *factory method pattern* dan yang tidak menerapkan *factory method pattern*.
5. Penyusunan laporan  
Hasil penelitian akan disusun menjadi suatu laporan yang meliputi aspek-aspek dalam penelitian yaitu teori dan implementasinya.