

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. Para *programmer* cenderung menyelesaikan masalah yang ada, dengan membuat desain perangkat lunak yang ada[13].

Dalam perkembangannya, banyak masalah-masalah yang kerap terjadi dalam pembangunan aplikasi berorientasi objek. Kebanyakan masalah tersebut memiliki inti permasalahan yang sama. Solusi umum dari permasalahan yang kerap muncul tersebut telah diselesaikan oleh *developer* sebelumnya, yang disebut *design pattern*.

Salah satu permasalahan yang kerap muncul adalah terdapatnya beberapa kondisi (*state*) pada perangkat lunak yang akan kita buat. Perpindahan tiap *state* akibat adanya suatu *event* harus benar-benar dapat ditanggulangi dengan baik oleh sebuah desain perangkat lunak. Permasalahan seperti itu dapat diselesaikan dengan *state pattern*, yang akan menjamin perpindahan setiap *state* yang ada sesuai dengan *state diagram* yang ada.

Dalam sebuah jasa travel, memerlukan adanya system pencatatan penumpang, untuk memudahkan proses bisnisnya. Untuk membuat aplikasi kasus tersebut, mempunyai permasalahan yang sama yang dapat diselesaikan menggunakan *state pattern*, karena aplikasi tersebut memiliki beberapa kondisi (status) seperti kosong, terisi dan penuh.

Object-oriented metrics (OO metrics) dapat digunakan untuk menganalisis perangkat lunak yang menerapkan konsep pemrograman berorientasi objek. Dengan parameter yang ada, *OO metrics* dapat menilai *reusability*, *complexity*, *coupling* dan *cohesion* sebuah perangkat lunak.

1.2 Perumusan masalah

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

1. Bagaimana menyelesaikan sebuah kasus dengan menggunakan *state pattern*?
2. Bagaimana menganalisis sebuah perangkat lunak yang menerapkan *state pattern*?
3. Apakah kelebihan dan kekurangan perangkat lunak yang menerapkan *state pattern*?

Batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Requirement dari perangkat lunak ini adalah meliputi :
 - Lihat Penumpang
 - Tambah Penumpang
 - Kurangi Penumpang
 - Hapus Semua Penumpang

2. *Object-oriented metrics* yang digunakan untuk mengukur perangkat lunak adalah CBO, LCOM, WMC, dan DIT.
3. Pada perangkat lunak ini akan digunakan basisdata dengan menggunakan MySQL sebagai databasenya. Tetapi basisdata tersebut bukan merupakan fokus dari tugas akhir ini.
4. Analisis dan desain perangkat lunak dimodelkan dengan menggunakan Eclipse SDK 3.2.0.
5. Menggunakan bahasa pemrograman Java dan NetBeans 6.7 sebagai aplikasi GUI nya serta Microsoft Windows XP sebagai system operasinya.

1.3 Tujuan

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

- a) Terciptanya sebuah perangkat lunak yang menerapkan state pattern.
- b) Terperolehnya hasil analisis, berupa nilai-nilai pada setiap OO metrics yg digunakan untuk menganalisis state pattern.
- c) Terperolehnya kelebihan dan kekurangan state pattern berdasarkan OO metrics yang dibandingkan dengan perangkat lunak yang tidak menerapkan state pattern.

1.4 Metodologi penyelesaian masalah

1. Studi Literatur :
Hal ini meliputi pencarian referensi yang berkaitan dengan *state pattern*, implementasi serta karakteristik dari *state pattern*, *object-oriented metrics*, dan penerapan *state pattern* pada kode perangkat lunak, serta cara perhitungan *object-oriented metrics* pada sebuah perangkat lunak.
2. Analisis dan Desain :
Menganalisa dan merancang desain untuk membangun perangkat lunak baik yang menggunakan *state pattern* maupun yang tidak menerapkan *state pattern*.
3. Implementasi sistem:
Membangun perangkat lunak yang menerapkan state pattern dan yang tidak menerapkan state pattern.
4. Pengujian dan pengukuran:
Pengujian hasil implementasi perangkat lunak secara keseluruhan yaitu pengujian seluruh fungsionalitas perangkat lunak yang dibuat serta pengukuran *object-oriented metrics* terhadap perangkat lunak yang dibangun.
5. Analisis hasil pengukuran:
Menganalisis hasil pengukuran *object-oriented metrics*