

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia rekayasa perangkat lunak sangat pesat. Salah satu pengembangan dilakukan pada sebuah paradigma membangun sistem berdasarkan objek, yang memandang sebuah domain masalah sebagai sekumpulan objek yang memiliki atribut dan behaviour yang unik. Ada sebuah metode dalam pembangunan suatu perangkat lunak, metode ini berusaha untuk meminimalkan *error* yang terdapat pada perangkat lunak, metode ini dinamakan *Design-by-contract*(DBC).

Metode DBC ini mengharuskan setiap eksekusi memiliki *precondition* yang harus dipenuhi sebelum eksekusi dan *postcondition* yang harus dipenuhi setelah eksekusi. Terdapat kekurangan pada metode ini, yaitu tidak seluruh bahasa pemrograman dirancang untuk mendukung teori ini, walaupun ada beberapa pustaka tambahan pada beberapa bahasa pemrograman untuk mendukung teori ini. Selain itu, penggunaan DBC ini tercampur dengan kode proses bisnis, sehingga akan menyulitkan untuk proses pemeriksaan kode sumber bila sistem yang dibangun cukup besar dan sewaktu pembangunan sistem, banyak fungsi yang mengimplementasikan DBC.

AOP(*aspect oriented programming*) menangani *crosscutting concern* yang merupakan suatu metode yang menangani suatu sifat tertentu yang tersebar dalam beberapa objek berbeda. Contohnya pada aspek *logging* atau *security* yang biasanya terdapat pada lebih dari satu definisi objek pada perangkat lunak. DBC akan diimplementasikan menggunakan AOP dan *assertion* akan ditangani sebagai *crosscutting concerns*.

Java merupakan bahasa pemrograman berorientasi-objek. Bahasa ini merupakan bahasa yang dapat berjalan pada lebih dari satu *platform*. Bahasa java tidak diciptakan khusus untuk menangani metode DBC.

1.2 Perumusan Masalah

Tugas akhir ini akan diitikberatkan pada:

- Bagaimana menerapkan metode DBC menggunakan AOP pada bahasa Java.
- Bagaimana menghasilkan perangkat lunak dengan *error* yang minimal dalam menangani *bug* dalam program, yaitu bila terjadi *error*, perangkat lunak akan memberikan sebuah peringatan.

Setelah implementasi DBC menggunakan AOP dilakukan, hasil dari implementasi tersebut akan dibandingkan dengan implementasi DBC menggunakan fitur yang ada pada Java, yaitu dengan menggunakan *keywords assert*.

Ruang lingkup dari tugas akhir ini adalah:

- Tugas akhir ini hanya akan menitikberatkan bagaimana metode DBC dapat diterapkan dengan menggunakan metode AOP dengan bahasa pemrograman Java.
- *Error* pada tugas akhir ini merupakan *error* yang telah diprediksi oleh pembuat program.
- Pengecekan *assertions* dilakukan dengan menggunakan kondisional *if-then-else* saja.
- Tugas akhir ini tidak akan membahas implementasi DBC menggunakan AOP pada kode-kode program yang berhubungan dengan antarmuka pemakai.
- Tugas akhir ini tidak membahas *reusability* dari *assertions*.
- Tugas akhir ini tidak akan membahas mengenai penurunan *assertions*.
- *Invariant* dalam tugas akhir ini tidak termasuk *loop invariant*.
- Penggunaan fitur *assert* tidak akan diimplementasi menggunakan AOP.
- Penggunaan fitur *assert* tidak akan menjadi bahasan utama dan tidak dibahas secara mendalam dalam tugas akhir ini.
- Aplikasi yang dibangun hanya akan menangani:
 - Entri CD(*Compact Disc*) maupun DVD(*Digital Video Disc*)
 - Entri Film baru.
 - Entri peminjam baru.
 - Peminjaman CD dan film DVD.
 - Proses pengembalian CD maupun film DVD.
- Pengguna hanya ada satu orang yaitu pemilik dari CD dan DVD tersebut.
- Untuk mendukung aplikasi, digunakan basis data yang menyimpan data-data CD, DVD, peminjam dan transaksi peminjaman itu sendiri.
- Analisis dan desain perangkat lunak akan dimodelkan dalam UML dengan menggunakan Rational Rose.
- Bahasa pemrograman yang digunakan adalah Java dengan menggunakan Borland JBuilder 9 sebagai IDE dan AspectJ sebagai modul tambahan untuk implementasi pemrograman Berorientasi-Aspek.
- Database yang digunakan adalah MySQL.
- Sistem Operasi yang digunakan adalah Microsoft Windows XP.

1.3 Tujuan Pembahasan

Tujuan tugas akhir ini adalah:

- Menerapkan DBC menggunakan metode pemrograman Berorientasi-Aspek dengan bahasa pemrograman Java.
- Membuat perangkat lunak yang dapat menangani *error*, bila pada saat pengujian terjadi *error*, perangkat lunak dapat menampilkan pesan yang relevan.
- Menganalisis dengan membandingkan hasil implementasi DBC menggunakan AOP dengan implementasi DBC menggunakan fitur *assert*.

1.4 Metodologi Pemecahan Masalah

Metode yang akan digunakan untuk menyelesaikan tugas akhir ini adalah :

- Studi literatur, yaitu dengan mempelajari literatur-literatur yang relevan dengan permasalahan, meliputi studi pustaka dan pencarian referensi mengenai DBC, AOP, dan Java.
- Mendefinisikan langkah-langkah implementasi DBC menggunakan AOP.
- Melakukan analisis dengan membandingkan antara implementasi DBC menggunakan AOP dan implementasi DBC menggunakan fitur *assert*.
- Pengambilan kesimpulan dan penyusunan laporan.