

ANALISIS DAN IMPLEMENTASI PEMROGRAMAN BERORIENTASIATRIBUT DI EJB (STUDI KASUS : PERANGKAT LUNAK Pencarian Text dalam File)

Muhamad Heriyana¹, Dana Sulistyio Kusumo², Kusuma Ayu Laksitowening³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Salah satu teknik pemrograman yang kini populer adalah pemrograman Berorientasi-Objek yang membungkus atribut dan operasi menjadi unit modularitas yang disebut objek. Salah satu bahasa pemrograman yang berbasiskan kepada orientasi objek yaitu java. Java sudah banyak digunakan dalam pembuatan suatu perangkat lunak, mulai aplikasi biasa hingga sebuah aplikasi enterprise. Salah satu teknologi java dalam membuat aplikasi enterprise yaitu J2EE yang membutuhkan container salah satu nya EJB. Namun beberapa developer merasa API dari EJB mempunyai spesifikasi yang kompleks.

Pada tugas akhir ini akan mencoba menerapkan pemrograman berorientasi attribute, sehingga akan memudahkan dalam proses pembangunan dan pemeliharaan dari suatu perangkat lunak. Perangkat lunak yang akan dibangun menggunakan EJB 2.1 dari pendekatan orientasi objek. Dan dari pendekatan orientasi attribut menggunakan XDoclet dan EJB 3.0. Berdasarkan pendekatan ini, kita dapat melihat bahwa proses pembangunan dengan menggunakan attribut dapat mengurangi kompleksitas menulis kode program dan memiliki nilai maintain lebih baik.

Kata Kunci : Objek, attribut, EJB, XDoclet, Maintainabilty

Abstract

One of programming technique that popular nowadays is Object-Oriented programming which encapsulates attributes and operations into single unit called object. Java is one of the programming language which is object oriented. Java is already been used in many software building and development, general application development to enterprise application development. J2EE is one of many technologies that java use to build an application, that need a container such as EJB. Though some developer perceive that the API from EJB has complex specification.

In this final task, i will try to implement the attribute oriented programming, so that the software development process and the software maintenance process will be easier. The software is developed by using EJB 2.1 with object oriented approach, and also with attribute oriented approach using XDoclet and EJB 3.0. Based on these approaches, we could see that development process using the attribute could reduce the complexity of writing code program and has a better maintain value.

Keywords : object, attribute, EJB, XDoclet, maintainability

1. PENDAHULUAN

1.1 Latar belakang

Dunia rekayasa perangkat lunak berkembang sangat pesat, termasuk dalam teknik pemrograman. Dari pemrograman yang tiap barisnya diinterpretasikan secara sekuens dan hampir tidak mengenal modularitas, hingga pemrograman prosedural dan kemudian pemrograman yang berorientasi pada objek. Peningkatan teknologi memungkinkan para pengembang perangkat lunak untuk memandang suatu domain masalah secara lebih sederhana, yaitu dengan adanya pemisahan bagian-bagian tertentu dari perangkat lunak sehingga lebih mudah diidentifikasi dan dimanipulasi. Kebutuhan akan modularitas semacam ini kemudian menghantarkan pada pemrograman yang berorientasi pada Objek (*Object-Oriented programming*).

Salah satu bahasa pemrograman komputer yang berbasis kepada *Object-Oriented programming* adalah java, pada saat ini java sudah banyak digunakan untuk membuat suatu perangkat lunak dari mulai aplikasi biasa hingga sebuah aplikasi enterprise. Berbagai fungsi yang dibutuhkan untuk sebuah aplikasi enterprise lebih mahal jika dibandingkan dengan beberapa aplikasi yang biasa. Fungsi-fungsi pada aplikasi enterprise biasanya termasuk pengaksesan secara intensif terhadap berbagai variasi dari sumber data (*datasource*) dan berbagai variasi antarmuka pengguna (*user interface*) atau aplikasi dalam bentuk web maupun aplikasi *stand alone*.

J2EE atau Java2 Enterprise Edition adalah sebuah teknologi yang banyak digunakan dalam pengembangan aplikasi enterprise dewasa ini. Sedangkan aplikasi enterprise merupakan aplikasi yang kompleks dan membutuhkan banyak sumber daya (*resource*). Salah satu komponen yang mendukung pembuatan aplikasi enterprise dalam J2EE adalah Enterprise Java Bean (EJB), EJB menyediakan sebuah standarisasi untuk mengimplementasikan suatu pengkodean dalam pembuatan aplikasi enterprise.

EJB banyak diadopsi oleh perusahaan-perusahaan besar namun masalah-masalah pun mulai bermunculan sehingga reputasi dari EJB mulai menurun, beberapa developer merasa bahwa API dari EJB ternyata lebih kompleks dari beberapa yang pernah digunakan oleh para *developer*, seperti terlalu banyak *checked exception*, membutuhkan *interface-interface*, dan mengimplementasikan sebuah *bean class* sebagai *abstract class* yang sangat tidak umum bagi kebanyakan *developer*. Namun pada EJB versi 2.1 *developer* dibantu oleh Xdoclet dalam membuat suatu aplikasi yang menggunakan EJB tersebut dengan menggunakan suatu anotasi tag dari javadoc sehingga membantu developer menggenerate code-code yang sering di ulang. Anotasi-anotasi tersebut

merupakan suatu metode pemrograman berorientasi atribut. Dalam perkembangannya EJB versi 3.0 juga menggunakan metode pemrograman berorientasi atribut dengan menggunakan anotasi-anotasi untuk mengurangi suatu kompleksitas versi sebelumnya

1.2 Perumusan Masalah

Berdasarkan uraian pada latar belakang, tampak permasalahan timbul pada EJB 2.1 yang mempunyai spesifikasi standarisasi yang kompleks sehingga cukup rumit untuk di bangun dan di pelihara. Oleh karena itu dibutuhkan suatu teknik bagaimana mempermudah untuk membangun suatu perangkat lunak agar lebih sederhana dan mudah di pelihara yaitu dengan pemrograman berorientasi atribut.

Untuk mendapatkan gambaran yang cukup mengenai permasalahan yang timbul maka dalam tugas akhir ini akan digunakan studi kasus perangkat lunak pencarian text dalam file, dalam perangkat lunak ini akan terlihat kompleksitas dalam membangun perangkat lunak dengan menggunakan EJB 2.1 dan bagaimana pemrograman berorientasi atribut membantu mengurangi kompleksitas tersebut. Dalam hal ini yang menggunakan pemrograman berorientasi atribut adalah XDoclet dan perkembangan dari EJB sendiri yaitu EJB 3.0

1.3 Tujuan Pembahasan

Tujuan pembuatan tugas akhir ini adalah menerapkan pemrograman berorientasi atribut sebagai suatu cara untuk mengurangi kompleksitas pada EJB 2.1 sehingga lebih mudah di bangun dan dipelihara dengan menggunakan XDoclet sebagai *tools* tambahan, dan penyederhanaan yang dilakukan oleh EJB 3.0.

Setelah proses pembangunan, akan dianalisis usaha pembuatandan nilai *maintainability* dari perangkat lunak dengan menggunakan EJB 2.1 sebagai pendekatan OOP dan XDoclet dan EJB 3.0 sebagai pendekatan @OP.

1.4 Batasan Masalah

Untuk menghindari terlalu meluasnya pembahasan tugas akhir ini, maka terdapat batasan masalah sebagai berikut :

1. Bahasa pemrograman yang digunakan dalam pendekatan OOP dan @OP yaitu java bukan C#.
2. Framework yang digunakan dalam implementasi pendekatan @OP yaitu Enterprise Java Bean 3.0, dan Xdoclet, dan implementasi pendekatan OOP yaitu Enterprise Java Bean 2.1.
3. Tipe kontainer EJB yang digunakan Session Bean.
4. Fungsi-fungsi dalam studi kasus pencarian teks menggunakan library apache lucene.

5. File yang dicari berupa file berekstensi text (.txt).
6. Pencarian file hanya dilakukan di dalam satu folder dan tidak melakukan rekursif ke dalam subfolder.
7. Analisis dan desain perangkat lunak akan dimodelkan dalam UML.

1.5 Metodologi Penelitian Masalah

Metodologi yang akan digunakan dalam merealisasikan tujuan dan pemecahan masalah di atas adalah dengan menggunakan langkah-langkah berikut:

1. Studi literatur

Pada tahap ini mempelajari literatur-literatur yang relevan dengan permasalahan yang meliputi studi pustaka dan pencarian referensi tentang pemrograman berorientasi atribut, serta bahasa pemrograman java dan EJB sebagai framework yang digunakan. Dan juga mempelajari library-library pendukung lainnya seperti apache lucene sebagai library dalam kasus pencarian text dalam file.

2. Analisis dan Desain

Pada tahap ini dilakukan analisis dan desain dengan menggunakan metode berorientasi objek untuk studi kasus pencarian teks dalam file, bahasa pemodelan yang akan digunakan adalah UML (Unified Modeling Language).

3. Implementasi

Implementasi berdasarkan desain kemudian dihubungkan dengan kedua framework yang berbeda yang menggunakan metode *Attribute Oriented Programming*. Bahasa pemrograman yang akan dipakai adalah java dan framework yang digunakan adalah *Enterprise Java Bean (EJB)*.

4. Analisis hasil dan Pengujian

Menguji dan menganalisis kedua framework yang menggunakan metode *Attribute Oriented Programming*.

5. Penyusunan Laporan

Membuat dokumentasi dari hasil analisis dan penarikan kesimpulan.

1.6 Sistematika Penulisan

- BAB I** **PENDAHULUAN**
Berisi latar belakang, perumusan masalah, tujuan pembahasan, batasan masalah, metodologi penyelesaian masalah dan sistematika penulisan.
- BAB II** **LANDASAN TEORI**
Membahas teori dan konsep dasar mengenai pemrograman Berorientasi-Attribut dan studi kasus.
- BAB III** **ANALISIS DAN DESAIN PERANGKAT LUNAK**
Membahas tentang analisis dan desain perangkat lunak dengan pendekatan Berorientasi-Objek sebagai dasar penerapan pemrograman Berorientasi-Attribut, dilengkapi dengan library-library yang digunakan.
- BAB IV** **DESAIN, IMPLEMENTASI DAN ANALISIS PEMROGRAMAN BERORIENTASI-ATTRIBUT**
Membahas tentang implementasi pemrograman Berorientasi-Attribut dalam pembangunan perangkat lunak serta pembahasan lebih lanjut mengenai karakteristik pemrograman Berorientasi-Attribut secara umum.
- BAB V** **KESIMPULAN DAN SARAN**
Berisi kesimpulan akhir dan saran pengembangan lebih lanjut.

5. PENUTUP

5.1 Kesimpulan

Dari hasil implementasi dan pengujian yang dilakukan diambil kesimpulan sebagai berikut :

1. Pemrograman berorientasi atribut dapat mengurangi penulisan dari suatu kode program pada pendekatan object oriented sehingga akan mudah dibaca dan dipelihara, dapat dilihat dari analisa pada saat proses development dan effort dari halstead metric.
2. Dari nilai maintainability index dapat dilihat bahwa pemrograman berorientasi atribut mempunyai nilai maintainability yang lebih baik karena mengurangi ukuran dari suatu kode program dan membantu untuk memelihara program tersebut lebih baik.

5.2 Saran

1. Pemrograman berorientasi atribut dapat di implementasikan pada framework yang lainnya sehingga dapat mempermudah developer.
2. Selain dari segi kemudahan dan pemeliharaan suatu perangkat lunak pemrograman berorientasi attribute dapat di uji kualitas nya dari segi yang lain, dalam hal ini EJB 3.0 dan EJB 2.1 dapat di uji dari segi functionality atau efficiency dan lain-lain.

6. DAFTAR PUSTAKA

- [1] Apache Software Foundation. Apache Lucene. 2008.<http://lucene.apache.org>
- [2] Anderson, Magnus dan Patrik Vestergreen, *Object Oriented Design Quality Metrics*, Information Technology Computing Science Departement, Uppsala University, Sweden.
- [3] http://en.wikipedia.org/wiki/ISO_9126
- [4] Harold, Elliotte Rusty. 1999. *Java IO*. O'Reilly.
- [5] Harold, Elliotte Rusty. 2002. *Processing XML with Java*. Addison-Wesley.
- [6] [Ibm.com/developerWorks](http://ibm.com/developerWorks), *Enterprise JavaBeans Fundamentals*.
- [7] Oppedijk, Frank R. 25 July 2008. Comparison of the SIG Maintainability Model and the Maintainability Index. University of Amsterdam Faculty of Science
- [8] Pressman, Roger S. 2001. *Software Engineering: A Practitioner's Approach, Edisi Kelima*. New York: McGraw-Hill.
- [9] Rouvoy, Romain and Philippe Merle. *Leveraging Component-Oriented Programming with Attribute Oriented*
- [10] Schwarz, Don. 2004. *Peeking Inside the Box: Attribute-Oriented Programming with Java 1.5, Part I*.
- [11] Wada, Hiroshi and J. Suzuki. October 2005. *Modeling Turnpike Frontend System: a Model-Driven Development Framework Leveraging UML Metamodeling and Attribute-Oriented Programming*. In Proc. of the 8th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS/UML 2005), Montego Bay, Jamaica.
- [12] Wada, Hiroshi and Jun Suzuki. 2005. *An Introduction to Attribute-Oriented Programming*. Boston: University of Massachusetts, Department of Computer Science.
- [13] Walss, Craig and Norman Richards. 2004. *XDoclet In Action*. Greenwich: Bruce Park Avenue