# 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 berbasiskan 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 perusahan-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 annotasi tag dari javadoc sehingga membantu developer menggenerate code-code yang sering di ulang. Annotasi-annotasi tersebut

merupakan suata metode pemrograman berorientasi attribut. Dalam perkembangannya EJB versi 3.0 juga menggunakan metode pemrograman berorientasi attribut dengan menggunakan annotasi-annotasi 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 perang lunak agar lebih sederhana dan mudah di pelihara yaitu dengan pemrograman berorientasi attribut.

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 attribut membantu mengurangi kompleksitas tersebut. Dalam hal ini yang menggunakan pemrograman berorientasi attribut adalah XDoclet dan perkembangan dari EJB sendiri yaitu EJB 3.0

## 1.3 Tujuan Pembahasan

Tujuan pembuatan tugas akhir ini adalah menerapkan pemrograman berorientasi attribut 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 Metedologi 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 attribut, 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

   Implentasi 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 diguanakan adalah *Enterprise Java Bean* (EJB).

4. Analisis hasil dan Pengujian

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

5. Penuyusunan 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.