

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam ranah ilmu informatika mendeteksi kemiripan kode program sangat dibutuhkan. Salah satu fungsi aplikasi ini adalah untuk mendeteksi terjadinya praktik plagiarisme dalam lingkungan akademis pemrograman. Selain itu, dengan sedikit modifikasi, teknik ini dapat pula diterapkan untuk mendeteksi kemiripan antar bagian program untuk memudahkan proses modularisasi program yang sudah ada.

Pada perkuliahan Pemrograman Berorientasi Objek (PBO) beberapa kali ditemukan mahasiswa yang melakukan plagiarisme, terutama saat pelaksanaan *assessment* dan tugas. Namun pada saat pemeriksaan tidak bisa diketahui dengan pasti siapa yang menjadi sumber plagiat dan siapa yang memplagiat *source codenya*. Oleh karena itu diperlukan sistem untuk melakukan pendeteksian plagiarisme pada mata kuliah PBO.

Plagiarisme merupakan tindakan penipuan hasil karya orang lain tanpa sepengetahuan dari penulis aslinya, yang melanggar suatu Hak Cipta dan Hak Moral. Ketertarikan mahasiswa terhadap tindakan plagiarisme, dibangun oleh rasionalitas instrumental. Mahasiswa lebih memperhitungkan tentang efisiensi, efektifitas dan nilai yang dimiliki oleh sumber dayanya (tugas akademik) untuk mencapai tujuan yang diharapkan. Ketika aktor (mahasiswa) menentukan tujuan, aktor akan dihadapkan pada sebuah pilihan cara alternatif yaitu cara SKS (Sistem Kebut Semalam) dan SKJ (Sekali Kerja Jadi). Pilihan tersebut akan memunculkan suatu bentuk tindakan plagiarisme dan konsekuensi dari tindakan plagiarisme.[6]

Pada proyek akhir ini, akan dibuatkan aplikasi untuk mendeteksi kemiripan kode pada mata kuliah PBO. Kode yang akan dilakukan pengecekan kemiripannya adalah kode hasil *assessment* dan tugas mata kuliah PBO. Pendeteksian kemiripan kode pada mata kuliah PBO ini dilakukan agar memudahkan para dosen untuk mengoreksi jawaban dari mahasiswa dan agar memudahkan proses penilaian dari *assessment* dan tugas.

Untuk mendeteksi kemiripan kode pada mata kuliah Pemrograman Berbasis Objek, digunakan metode pengecekan kode menggunakan *Abstract Syntax Tree (AST)*. *Abstract Syntax Tree (AST)* merupakan representasi pohon struktur sintaksis abstrak kode yang ditulis dalam bahasa pemrograman. Setiap node dari pohon menunjukkan suatu konstruksi yang terjadi dalam kode.

1.2 Rumusan Masalah

Beberapa rumusan masalah yang akan dibahas dalam Proyek Akhir ini adalah sebagai berikut.

1. Bagaimana cara mendeteksi kemiripan kode pada hasil *assessment* dan tugas mata kuliah PBO?
2. Bagaimana cara memberikan penilaian pada kode yang sudah dicek?

1.3 Tujuan

Tujuan pembuatan Proyek Akhir ini adalah.

1. Membangun aplikasi untuk mendeteksi kemiripan kode program.
2. Membangun aplikasi yang dapat memberikan penilaian dari hasil pengecekan dengan menambahkan parameter penilaian.

1.4 Batasan Masalah

Batasan masalah dari Proyek Akhir ini adalah.

1. Kode program yang akan digunakan adalah kode program *assessment* pada mata kuliah Pemrograman Berbasis Objek (PBO) semester 3 pada program pendidikan manajemen informatika.
2. Deteksi kemiripan kode program akan menggunakan metode *Abstract Syntax Tree*.
3. Aplikasi ini hanya membandingkan dua sumber *source code*.
4. Aplikasi ini hanya dapat mendeteksi variabel dan metod dengan tipe data dasar yaitu *byte, short, int, long, float, double, char, boolean, String*.
5. Penilaian *source code* terbagi menjadi 4 bagian : nama kelas, variabel, metod dan fungsi.
6. Aplikasi ini tidak dapat mendeteksi apabila ada 2 deklarasi variabel dalam 1 baris.

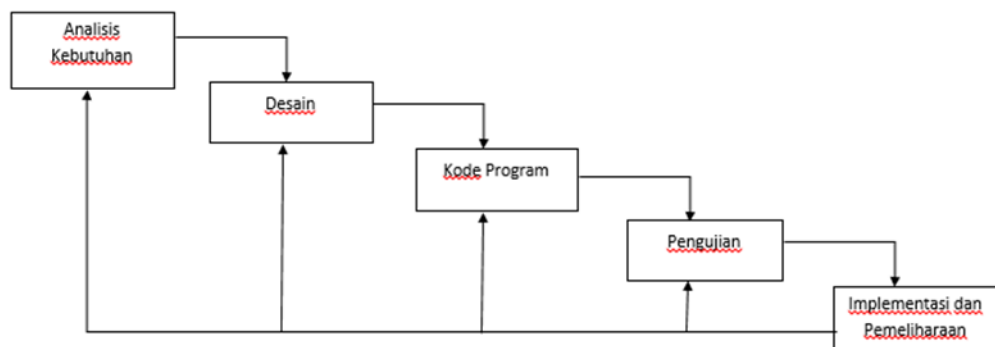
1.5 Definisi Operasional

Aplikasi plagiarisme adalah aplikasi berbasis *desktop* yang dibangun untuk membantu dosen dalam memeriksa *source code assessment* dan memberikan penilaian pada *source code*. Aplikasi ini akan digunakan untuk pengecekan *assessment* pada mata kuliah Pemrograman Berorientasi Objek (PBO). PBO adalah paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi dibungkus dalam kelas-kelas atau objek-objek. Untuk mendeteksi plagiarisme, aplikasi ini akan menggunakan metode *Abstract Syntax Tree* (AST). AST adalah representasi sebuah kode program kedalam pohon node dan setiap node menunjukkan algoritma dari kode program.

Dalam pembangunan aplikasi ini menggunakan bahasa pemrograman *Java*. Sedangkan untuk metode pengerjaan dalam pembangunan aplikasi ini adalah Prototype.

1.6 Metode Pengerjaan

Model pengerjaan Aplikasi Deteksi Plagiarisme Pada Mata Kuliah Pemrograman Berorientasi Objek ini menggunakan *Waterfall Model* oleh Pressman. Berikut tahapan dalam *Waterfall Model* meliputi. [11]



Gambar 1-1 Metode Waterfall

1. Analisis Kebutuhan

Pada tahap ini dilakukan pengumpulan kebutuhan dan analisis dengan cara wawancara ke dosen PBO untuk mengetahui proses seperti apa yang sedang berlangsung saat pengecekan kode *assessment* dan tugas. Pada tahap ini melibatkan entitas dosen sebagai pengguna aplikasi.

2. Desain

Setelah melakukan mendapatkan data dan melakukan analisis yang dibutuhkan dari tahap sebelumnya, selanjutnya adalah melakukan perancangan sistem yang akan berjalan pada aplikasi. Pada tahap ini

dilakukan perancangan berupa *mockup* dan *use case* dari aplikasi yang akan dibuat.

3. Kode Program

Pada tahap ini dilakukan pengkodean. Aplikasi ini akan dibuat dengan menggunakan bahasa pemrograman *Java*.

4. Pengujian

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, kemudian dilakukan proses pengujian sistem. Pengujian ini dilakukan dengan *white box*.

5. Implementasi dan Pemeliharaan

Setelah melalui beberapa tahapan diatas dan dipastikan bahwa program telah terbebas dari kesalahan dan mengalami perbaikan maka program diimplementasikan pada komputer pengguna.

1.7 Jadwal Pengerjaan

Jadwal pengerjaan proyek akhir ini sebagai berikut:

Tabel 1-1 Jadwal Pengerjaan

no	kegiatan	TAHUN 2017																							
		BULAN																							
		JANUARI				FEBRUARI				MARET				APRIL				MEI				JUNI			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Analisis Kebutuhan																								
2	Desain																								

[illegible]