

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Plagiarisme bisa menjadi masalah besar di universitas-universitas. Ada penelitian yang melaporkan bahwa lebih dari 10% siswa melakukan kasus plagiarisme [1]. Siswa dapat dengan mudah meniru tugas *programming* milik temannya dan memodifikasinya agar tidak terlihat seperti mencontek. Oleh karena itu penting untuk mengembangkan sistem deteksi plagiarisme yang secara otomatis menghitung kesamaan sepasang kode sumber.

Penelitian untuk mendeteksi plagiarisme sebagian besar dibagi menjadi dua kelompok, yaitu mendeteksi kasus plagiarisme di media dokumen teks yang ditulis dalam bahasa alami dan mendeteksi plagiarisme antara program komputer. Pada kenyataannya, untuk mendeteksi plagiarisme itu sebenarnya bukanlah tugas sebuah komputer akan tetapi itu tugas manusia, karena untuk mendeteksi plagiarisme itu sebenarnya dibutuhkan pemahaman dari dokumen tersebut.

Namun untuk manusia, memahami kode sumber program komputer itu jauh lebih sulit daripada dokumen normal. Jadi sistem deteksi otomatis diperlukan untuk membandingkan kode sumber program komputer. Ada beberapa penelitian untuk mendeteksi kesamaan antara program komputer, seperti MOSS yang menggunakan metode *Winnowing* [2], YAP3 menggunakan *Rabin-Karp Greedy-String-Tiling* [3], JPlag menggunakan *Greedy-String-Tiling* [4].

Sebagian besar detektor ini membandingkan kode sumber program komputer secara sintaksis. Namun tidak semua potongan kode sumber dengan makna yang sama memiliki bentuk sintaks yang serupa, seperti baris instruksi *for* dan *while*. Secara sintaks berbeda, akan tetapi secara makna mirip. Mengingat permasalahan tersebut maka akan dibangun sebuah detektor yang membandingkan kode sumber secara semantik. Perbandingan ini dilakukan dengan mentranslasikan kode sumber menjadi (*Common Intermediate Language*) yang disingkat menjadi

CIL. CIL adalah sebuah *intermediate language* yang digunakan oleh .NET dan Mono Framework.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan maka rumusan masalah yang dibahas dalam penelitian ini adalah :

1. Bagaimana mendeteksi kesamaan program secara otomatis pada mata kuliah pemrograman dasar?
2. Bagaimana membandingkan kesamaan kode sumber secara efektif dan efisien?

## 1.3 Tujuan Pembahasan

Berdasarkan rumusan masalah yang telah dijelaskan maka tujuan pembahasan yang dibahas dalam penelitian ini adalah :

1. Membuat aplikasi yang dapat mendeteksi kesamaan program secara semantik.
2. membandingkan kode sumber dalam pendekatan *intermediate language*. Pada pendekatan tersebut, token delimiter akan terbuang sehingga lebih sedikit, dan hanya memperhitungkan token semantik.

## 1.4 Ruang Lingkup

Ruang lingkup kajian dari pemaparan masalah meliputi :

1. Kode sumber yang bisa digunakan adalah semua kode berbasis .NET dan dalam pengujiannya menggunakan bahasa C#.
2. Kode sumber yang tidak lulus tahap kompilasi akan langsung dibandingkan secara sintaksis.
3. Membandingkan kode sumber menggunakan algoritma *Adaptive Local Alignment*.
4. Deteksi kesamaan kode sumber terfokus pada kode sumber yang berbasis *problem solving*, dimana program terdiri dari input, proses dan output.

## 1.5 Sumber Data

Perancangan aplikasi ini menggunakan 2 sumber data, yaitu sumber data primer dan sekunder. Data primer didapatkan adalah data kode sumber dataset

pengujian yang dipakai dalam jurnal berjudul *Detecting Source Code Plagiarism on Introductory Programming Course Assignments Using a Bytecode Approach* [5] yang memakai konsep 6 level plagiarisme dalam jurnal *An empirical approach for detecting program similarity and plagiarism within a university programming environment* [6]. Untuk sumber data sekunder didapat dari jurnal penelitian yang terkait dengan topik algoritma local alignment, dan topik pendekatan *intermediate language*.

## **1.6 Sistematika Penyajian**

### **BAB 1 PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, tujuan pembahasan, ruang lingkup bahasan, sumber data, sistematika penyajian, dan waktu dan tempat penelitian.

### **BAB 2 KAJIAN TEORI**

Bab ini membahas mengenai teori-teori yang berkaitan dalam penyelesaian proyek tugas akhir ini.

### **BAB 3 ANALISIS DAN RANCANGAN SISTEM**

Bab ini membahas mengenai analisis dan rancangan sistem yang terdapat di tugas akhir ini, dan juga rancangan antarmuka dari proyek yang dikerjakan dalam tugas akhir ini.

### **BAB 4 IMPLEMENTASI**

Bab ini berisi kumpulan *screenshot* dari proyek yang dibuat beserta penjelasan dari tiap fitur utama yang dibuat.

### **BAB 5 PENGUJIAN**

Bab ini berisi pembahasan dan uji coba program dari fitur utama yang telah dibuat.

### **BAB 6 KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan pembahasan pada perancangan analisa pengujian aplikasi yang dibuat, serta saran untuk menyempurnakan dan memperbaiki aplikasi supaya menjadi aplikasi yang lebih baik.