

1. Pendahuluan

1.1 Latar Belakang Masalah

Plagiarisme adalah menggunakan hasil kerja orang lain tanpa memberi penghargaan pada mereka [2]. Penghargaan di sini berarti mencantumkan nama orang lain tersebut. Jika ingin mencantumkan kata atau ide orang lain, masukkan dalam daftar pustaka dan sebutkan sumbernya. Walaupun kata atau ide tersebut sudah diterjemahkan ke dalam kata-kata kita sendiri.

Plagiarisme dapat dilakukan di mana saja. Salah satunya di lingkungan yang dekat dengan teknologi informasi [17]. Misalnya di lingkungan kampus jurusan teknologi informasi maupun informatika. Praktik plagiarisme dilakukan dengan cara menyalin dan memodifikasi *source code* atau kode program yang dilakukan ketika mahasiswa harus menyelesaikan tugas yang diberikan oleh dosen. Modifikasi yang biasa dilakukan antara lain mengubah komentar, mengubah karakter *white space* dan *layout* program, mengganti nama *identifier*, mengubah urutan blok kode program, mengubah urutan *statement* dalam blok kode program, mengubah urutan operator pada *expression*, mengubah tipe data, menambahkan *variable* dan *statement* yang tidak perlu, mengganti *control structures* dengan *structure* lainnya yang *equivalent*, dan memasukkan fungsi maupun prosedur ke dalam program utama [11].

Ada dua cara untuk meminimalisasi plagiarisme ini, yaitu dengan mencegah dan mendeteksi. Mencegah berarti menghalangi dan menjaga agar plagiarisme tidak dilaksanakan [14]. Salah satu caranya adalah dengan memberikan peraturan yang jelas dan sanksi yang jelas apabila terbukti melakukan plagiarisme.

Sedangkan mendeteksi berarti melakukan pendeteksian secara *manual* terhadap tugas yang dikumpulkan. Kelemahan proses ini adalah waktu yang digunakan untuk melakukan pendeteksian, karena jumlah mahasiswa yang sangat banyak dan juga tingkat kreatifitas mahasiswa yang tinggi dalam melakukan manipulasi *source code* tersebut. Oleh karena itu perlu dibuat proses pendeteksian plagiarisme *source code* secara cepat dan tepat [4].

Pada Tugas Akhir ini, akan menganalisa dan mengimplementasikan algoritma *edit distance* untuk mengetahui kemiripan antara dua buah *source code*. algoritma *edit distance* merupakan masalah klasik di bidang ilmu komputer pada umumnya, dan di bidang pencocokan pola kombinasi pada khususnya [5]. Algoritma *edit distance* dapat menghitung perbedaan antara dua *string* berdasarkan operasi sisip, ubah, dan hapus dengan memanfaatkan kelebihan program dinamis yaitu solusi dibangun dari solusi pada tahap sebelumnya sehingga kompleksitas yang didapat jauh lebih baik dibandingkan algoritma *brute force* [6]. Kompleksitas waktu algoritma *edit distance* ini adalah $O(|String1| * |String2|)$ atau kuadratik $O(n^2)$ jika panjang kedua string sama. Algoritma pemrograman dinamis ini lebih baik daripada algoritma *brute force* dengan kompleksitas waktu eksponensial $O(3^n)$. Jadi, dapat disimpulkan bahwa algoritma ini sangat baik untuk diterapkan dalam pendeteksian praktik plagiat [4].

Pada Tugas Akhir sebelumnya yang menggunakan algoritma *edit distance*, *source code* yang dibandingkan masih dalam bentuk *string* [15]. Kemiripan yang bisa terdeteksi hanya dalam bentuk penambahan, pengurangan dan pemindahan baris program. Pada tingkat *string*, perubahan letak baris program belum bisa diketahui pengaruhnya untuk alur program maupun hasil akhirnya. Kemiripan *variable* antara

dua buah *source code* juga masih diidentifikasi berdasarkan namanya, bukan berdasarkan fungsi yang akan digunakan oleh *variable*.

Pada Tugas Akhir ini, algoritma *edit distance* digunakan pada tingkat *token*. *Source code* yang diinputkan oleh *user* dipisah berdasarkan *token-token*nya, dan kemudian *token-token* yang dihasilkan diidentifikasi berdasarkan fungsi-fungsi dan atributnya. Selanjutnya algoritma *edit distance* akan melakukan pengecekan kemiripan antara dua buah *source code* berdasarkan fungsi dan atribut yang ada di dalam *token-token* yang telah dihasilkan.

1.2 Perumusan Masalah

Dari latar belakang yang telah dijelaskan sebelumnya, maka permasalahan yang dibahas dan diteliti adalah:

1. Bagaimana menerapkan algoritma *edit distance* di tingkat *token* untuk membangun aplikasi pendeteksi plagiarisme *source code*.
2. Bagaimana akurasi algoritma *edit distance* untuk diimplementasikan pada aplikasi pendeteksi plagiarisme *source code*.
3. Faktor apa yang menentukan akurasi dari aplikasi pendeteksi plagiarisme *source code*.

Adapun yang menjadi batasan masalah pada Tugas Akhir ini adalah dibatasi hanya pada bahasa pemrograman tertentu dengan fungsi dan prosedur tertentu yang terdapat di dalam bahasa pemrograman tersebut.

1.3 Ruang Lingkup Masalah

Adapun yang menjadi ruang lingkup masalah pada Tugas Akhir ini adalah:

1. Bahasa pemrograman yang diuji adalah bahasa *Pascal*.
2. Bagian dalam *Pascal* yang digunakan dibatasi pada judul program, pendeklarasian *variable*, pendeklarasian *procedure*, pendeklarasian *function*, *assignment*, pemanggilan *procedure*, pemanggilan *function*, *while-do*, *repeat-until*, *for-to-do*, *case-of*, *if-then-else* serta *input-output*.
3. Modifikasi yang digunakan dibatasi hanya pada modifikasi komentar, modifikasi *identifier*, modifikasi *variable*, modifikasi *procedure* dan *function*, modifikasi *statement* dan modifikasi *control logic*.
4. Parser yang digunakan tidak mencantumkan bagian pendeklarasian yang digunakan untuk meng-include-kan file lain untuk dianggap sebagai satu program. Sehingga jika ada file lain yang seharusnya menjadi bagian dari program utama, maka file tambahan tersebut akan dianggap sebagai program lain.

1.4 Tujuan

Tujuan penyusunan Tugas Akhir ini adalah:

- a. Mengimplementasikan algoritma penyederhanaan *source code* untuk mengubah *source code* menjadi bentuk yang lebih sederhana.
- b. Mengimplementasikan dan menganalisis algoritma *edit distance* dalam pendeteksian plagiarisme *source code*.
- c. Menghasilkan suatu sistem pendeteksian plagiarisme antara dua buah *source code* dan menampilkan hasilnya dalam bentuk persentase.

1.5 Metodologi Penyelesaian Masalah

Dalam menyusun Tugas Akhir ini, digunakan beberapa metode untuk mendapatkan data, antara lain :

a. Studi Literatur

Metode ini dilaksanakan dengan melakukan studi literatur. Literature yang didapat dalam menyelesaikan tugas akhir ini berasal dari buku, *e-book*, *paper* dan artikel-artikel *hardcopy* maupun *softcopy*. Literatur yang menjadi bahan utama adalah literatur tentang teknik kompilasi, *edit distance* dan juga desain analisis dan algoritma. Literatur yang menjadi tambahan antara lain tentang bahasa yang akan digunakan.

b. Analisis dan Desain

Pada perancangan implementasi pendeteksi plagiarisme *source code* akan digunakan algoritma *edit distance*. Dan pada tahap ini dilakukan analisis terhadap algoritma *edit distance* serta dilakukan pula analisis kebutuhan perangkat lunak dan perancangan dari perangkat lunak.

c. Implementasi

Metode ini dilaksanakan dengan melakukan perancangan aplikasi yang akan dibangun sebagai pendeteksi plagiarisme dan mengimplementasikan algoritma *edit distance*. Agar algoritma *edit distance* bisa diaplikasikan di tingkat token, maka perlu dibangun sebuah *scanner* dan *parser*. Pada Tugas Akhir ini, *parser* juga berfungsi untuk memberikan informasi yang dibutuhkan untuk proses pendeteksian.

d. Pengujian dan Analisa Hasil

Metode ini dilaksanakan dengan melakukan pengujian terhadap perangkat lunak yang telah dibangun dan sekaligus melakukan analisis terhadap perangkat lunak. Output dari perangkat lunak ini adalah presentasi kemiripan dari satu *source code* terhadap sekumpulan *source code* yang lain.

e. Pembuatan Laporan

Pada tahap ini dilakukan penyusunan laporan hasil penelitian yang telah dilakukan dan membuat kesimpulan dari hasil penelitian tersebut.

Laporan yang disusun terdiri dari:

1. Pendahuluan

Pada bab ini akan menjelaskan mengenai latar belakang penelitian, permasalahan yang diteliti, tujuan penelitian dan metodologi penyelesaian masalah.

2. Landasan Teori

Landasan teori akan memberikan dasar pengetahuan yang dapat membantu penelitian.

3. Rancangan dan Gambaran Sistem

Pada tahap ini akan dilakukan perancangan dan bagaimana sistem akan diimplementasikan.

4. Pengujian dan Analisis

Pada tahap ini akan dilakukan pengujian dan analisa terhadap sistem yang telah dibangun. Dan berisi tentang hasil-hasil penelitian dan pengujian.

5. Penutup

Pada bab ini akan memberikan kesimpulan berdasarkan penelitian yang telah dikerjakan. Serta diberikan saran-saran yang dapat membantu pelaksanaan penelitian berikutnya yang mungkin dilakukan.

1.6 Jadwal Kegiatan

Jadwal pelaksanaan Tugas Akhir ini adalah sebagai berikut:

Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4				Bulan 5				Bulan 6				Bulan 7			
	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	2	3	4
Studi Literatur																												
Analisis dan Desain																												
Implementasi																												
Pengujian																												
Pembuatan Laporan																												