



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB I PENDAHULUAN

1.1 Latar Belakang

Plagiarisme menurut KBBI memiliki arti penjiplakan yang melanggar hak cipta. Menurut Sulianta (2007) dalam bukunya yang berjudul “Seri Praktis Konten Internet”, menjelaskan bahwa kegiatan plagiarisme marak terjadi di kalangan mahasiswa. Hal ini dipertegas dengan hasil survei yang dilakukan oleh Pew Research Center di tahun 2011 yang menunjukkan bahwa sekitar 55% rektor dari berbagai kampus di Amerika, menyatakan bahwa tindakan plagiarisme telah meningkat selama 10 tahun ini.

Penelitian berupa pemeriksaan plagiarisme pada sebuah *source code* berbahasa C pernah dilakukan sebelumnya oleh Andreas Arifianto dari Universitas Multimedia Nusantara pada tahun 2011 yang berjudul “Rancang Bangun Aplikasi Pendeteksi Plagiarisme Kode Program Bahasa C Menggunakan algoritma Levenshtein Distance dan Brute Force”. Dalam penelitian tersebut dilakukan perbandingan kinerja antara algoritma *levenshtein distance* dan algoritma *brute force* dalam mendeteksi plagiarisme. Penelitian tersebut menunjukkan bahwa algoritma *levenshtein* menghasilkan bobot persentase kecenderungan plagiarisme lebih tinggi pada saat membandingkan kode program yang mengalami perubahan yang bersifat non-struktural, sedangkan algoritma *brute force* menghasilkan bobot persentase kecenderungan plagiarisme lebih tinggi pada saat membandingkan dokumen kode program struktural. Berbeda dengan penelitian yang dilakukan oleh Regina Natalia dari Universitas

Multimedia Nusantara tahun 2011 dengan judul “Peningkatan Kemampuan Aplikasi Pendeteksi Plagiarisme pada Program Berbahasa Pemrograman C dengan Algoritma Boyer Moore dan Brute Force”. Dalam penelitian yang dilakukan Regina, dua algoritma yang dibandingkan ialah algoritma *brute force* dengan algoritma *Boyer Moore* yang menghasilkan kesimpulan bahwa algoritma *boyer moore* memiliki persentase lebih tinggi untuk kasus manipulasi kode program dengan perubahan struktural dan non-struktural.

Pada dasarnya algoritma *levenshtein distance* dan algoritma *hamming distance* memiliki persamaan, sebab keduanya termasuk algoritma *edit distance*. Menurut Lovrencic (2009), algoritma *edit distance* merupakan algoritma yang digunakan untuk mencari perbedaan atau selisih antar dua objek. Objek tersebut bisa berupa biner, *string* dan lainnya. Perbedaan algoritma *levenshtein distance* dan algoritma *hamming distance* terletak pada proses penyelesaian masalahnya. Berbeda dengan algoritma *levenshtein*, algoritma *hamming distance* hanya dapat bekerja pada dua buah objek yang memiliki ukuran atau panjang yang sama. Menurut Abdeen (2011), algoritma *brute force* adalah algoritma paling sederhana yang dapat digunakan pada pencarian pola. Sedangkan menurut Sinapova (2014), beberapa algoritma *brute force* yang terkenal pada kasus *search and sort* adalah *sequential search*, *selection sort*, dan *bubble sort*.

Penggunaan algoritma *hamming distance* dan *brute force* untuk mendeteksi tindakan plagiat memungkinkan terbentuknya aplikasi deteksi kemiripan *source code* dengan konsep yang sederhana.

1.2 Rumusan Masalah

Rumusan masalah penelitian ini adalah bagaimana mengimplementasikan algoritma *hamming distance* dan algoritma *brute force* dalam mendeteksi plagiarisme pada *source code* berbahasa C?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengimplementasikan algoritma *hamming distance* dan algoritma *brute force* dalam mendeteksi plagiarisme pada *source code* berbahasa C.

1.4 Batasan Masalah

Batasan dari penelitian ini adalah sebagai berikut.

1. *Source code* yang dikenali dalam bahasa C.
2. Input berupa dua buah *file* berekstensi “.c”.
3. Aplikasi berbasis *desktop* dengan bahasa C#.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah membantu pengguna khususnya kalangan pengajar pemrograman bahasa C dalam menghitung tingkat kesamaan dua buah *source code* berbahasa C.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada skripsi ini adalah sebagai berikut.

BAB I PENDAHULUAN

Pada bab ini, berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan skripsi ini.

BAB II LANDASAN TEORI

Bab ini mendeskripsikan tentang teori-teori ataupun metode yang berkaitan dengan penelitian yang dilakukan. Teori ataupun metode yang digunakan antara lain plagiarisme, algoritma *hamming distance*, algoritma *brute force*, dan *preprocessing*.

BAB III METODOLOGI PENELITIAN DAN PERANCANGAN

Bab ini berisikan metode penelitian dan rancangan aplikasi pendeteksi plagiarisme *source code* berbahasa C, baik rancangan fungsional maupun antarmuka pengguna.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan pengujian fungsional dari aplikasi pendeteksi plagiarisme *source code* berbahasa C. Selain itu, pada bab ini juga terdapat hasil pengujian yang telah dilakukan.

BAB V KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan dari penelitian yang dilakukan dan berisi saran yang diberikan untuk penelitian selanjutnya.

