



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

Perkembangan Teknologi dan Informasi telah berkembang sedemikian hebatnya. Kemajuan di bidang komputer dan internet khususnya, semakin mempercepat terjadinya perubahan yang sangat besar bagi manusia dalam berkomunikasi serta mencari dan bertukar informasi.

Jiménez-Peris, dkk dalam *paper*-nya yang berjudul “*New Technologies in Computer Science Education*” mengungkapkan bahwa metode pembelajaran dan pengajaran di masa yang akan datang akan sangat berubah. Dimana, perubahan itu disebabkan oleh kemajuan dalam bidang teknologi informasi dan komunikasi.

Selanjutnya, Jiménez-Peris, dkk juga mengungkapkan bahwa salah satu teknologi yang akan berkembang dalam dunia pendidikan adalah pendeteksi plagiarisme. Pendeteksi plagiarisme ini memungkinkan tenaga pengajar (dosen) untuk memeriksa adanya indikasi plagiarisme antara dua dokumen atau lebih. Sehingga waktu dan usaha yang dibutuhkan dapat lebih kecil dibanding dengan melakukan pemeriksaan secara manual.

Plagiarisme dalam dunia pendidikan sendiri sudah bukan menjadi hal yang baru. Dalam Kamus Besar Bahasa Indonesia, plagiarisme atau sering disebut plagiat didefinisikan sebagai penjiplakan atau pengambilan karangan,

pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri.

Sugiyanto (2010) mengungkapkan “Plagiarisme adalah suatu kejahatan akademik (*academic criminal*).... Plagiarisme adalah suatu ketidakadilan karena kepuasan atau manfaat yang diperoleh seseorang dilakukan dengan mengabaikan hak bahkan meniadakan hak orang lain untuk mendapatkan kepuasan atau manfaat yang sama.”

Secara khusus dalam cakupan mahasiswa fakultas ICT (*Information and Communication Technology*) di Universitas Multimedia Nusantara, cukup banyak ditemukan indikasi terjadinya praktik plagiarisme kode program. Berdasarkan wawancara yang dilakukan terhadap beberapa dosen dan asisten laboratorium untuk mata kuliah Algoritma dan Pemrograman, didapatkan fakta, bahwa cukup banyak terjadi tindak plagiarisme kode program yang dilakukan oleh mahasiswa. Dimana ditemukan rata-rata sekitar 25-30 kasus plagiarisme kode program untuk setiap kelasnya dalam 1 semester.

Secara umum, Lukashenko, dkk (2007, p.1) mengungkapkan ada dua cara untuk mengatasi permasalahan plagiarisme, yaitu dengan melakukan pencegahan dan pendeteksian. Mencegah dapat diartikan dengan berupaya sedini mungkin untuk menghalangi tindakan plagiat. Caranya bisa dimulai dengan motivasi untuk bekerja dengan kemampuan diri sendiri hingga pemberian sanksi atau efek jera bagi mahasiswa yang diketahui melakukan plagiat. Sedangkan mendeteksi berarti melakukan sesuatu untuk menemukan tindakan plagiat yang dilakukan.

Salah satu kesulitan yang dihadapi oleh para dosen dan asisten laboratorium dalam mendeteksi kecenderungan plagiarisme, berdasarkan hasil wawancara yang dilakukan adalah banyaknya dokumen kode program yang harus diperiksa, dimana waktu yang dimiliki sangat terbatas (5 suara). Selain itu, kesulitan lain yang dialami adalah ketelitian dalam membandingkan dokumen kode program (4 suara), konsep kelas paralel (3 suara), dan dokumentasi hasil pendeteksian dokumen kode program dalam jumlah yang cukup banyak (2 suara). Oleh karena itu, seluruh responden menyatakan bahwa diperlukan sebuah sistem yang terkomputerisasi yang dapat membantu mendeteksi kecenderungan plagiarisme antara dokumen kode program yang dibuat oleh mahasiswa.

Di samping itu, berbeda dengan aplikasi pendeteksi plagiarisme untuk dokumen teks dalam bentuk esai atau uraian, aplikasi pendeteksi plagiarisme dokumen kode program harus dibuat secara spesifik, sesuai dengan karakteristik dari kode program yang akan dibandingkan. Sehingga dibutuhkan algoritma yang dapat mendeteksi kecenderungan plagiarisme dokumen kode program dengan baik.

Algoritma *Levenshtein Distance* merupakan sebuah algoritma yang dapat menghasilkan metrik jarak antara dua *sequence* yang dibandingkan. Algoritma ini mampu menghitung jarak (perbedaan) antara dua *sequence (string)* berdasarkan operasi minimum yang diperlukan untuk merubah satu *string* menjadi *string* lainnya, melalui teknik *substitution*, *insertion*, dan *deletion*. Sehingga semakin banyak perbedaan diantara kedua dokumen kode program yang dibandingkan,

maka bobot persentase kecenderungan plagiarisme yang dihasilkan akan semakin kecil.

Namun, algoritma *Levenshtein Distance* memiliki kekurangan, dimana algoritma ini dapat ‘tertipu’ ketika membandingkan dokumen kode program yang dimanipulasi secara struktural (pertukaran posisi baris). Untuk itu, diperlukan sebuah algoritma tambahan yang dapat menghasilkan bobot persentase kecenderungan plagiarisme dengan konsep pencocokan baris per baris. Sehingga digunakan algoritma *Brute Force*, yang memungkinkan setiap baris pada dokumen yang memiliki baris terpendek akan dibandingkan dengan seluruh baris pada dokumen lainnya.

Berdasarkan latar belakang tersebut, penulis melakukan penelitian dengan judul “Rancang Bangun Aplikasi Pendeteksi Plagiarisme Kode Program Bahasa C Menggunakan Algoritma *Levenshtein Distance* Dan *Brute Force*”.

1.2 Rumusan Masalah

Masalah yang akan dikaji dalam penelitian ini adalah.

1. Bagaimana merancang dan membangun aplikasi pendeteksi plagiarisme yang cocok untuk kode program dalam bahasa C?
2. Bagaimana mengimplementasikan algoritma *Levenshtein Distance* dan *Brute Force* untuk membangun aplikasi pendeteksi plagiarisme kode program dalam bahasa C?

3. Bagaimana performa dari algoritma *Levenshtein Distance* dan *Brute Force* dalam mengidentifikasi bobot persentase kecenderungan plagiarisme antara dokumen kode program dalam bahasa C?
4. Bagaimana mengimplementasikan teknik *preprocessing* agar dapat meningkatkan bobot persentase kecenderungan plagiarisme yang dihasilkan oleh Algoritma *Levenshtein Distance* dan *Brute Force*?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah.

1. Ruang lingkup penelitian terbatas pada mahasiswa fakultas ICT (*Information and Communication Technology*) di Universitas Multimedia Nusantara untuk mata kuliah Algoritma dan Pemrograman.
2. Aplikasi hanya dapat membandingkan dokumen kode program dalam bahasa pemrograman C, dengan format *.c* (*C Source*) atau *.cpp* (*C++ Source*) yang bebas dari *syntax error*.
3. Aplikasi pendeteksi plagiarisme ini bekerja secara *offline* (*Intra-Corporal Detection*).
4. Cakupan materi kode program yang dibandingkan meliputi: *variable*, *conditional*, *looping*, *array*, *pointer*, *function*, dan *file*.
5. Bobot persentase yang dihasilkan sistem tidak memberikan keputusan mutlak terhadap dokumen yang diidentifikasi plagiat, melainkan hanya

sebagai panduan bagi pemakai aplikasi untuk menunjukkan kecenderungan plagiarisme antara dokumen-dokumen yang dibandingkan.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah.

1. Merancang dan membangun aplikasi pendeteksi plagiarisme yang cocok untuk kode program dalam bahasa C.
2. Mengimplementasikan algoritma *Levenshtein Distance* dan *Brute Force* untuk membangun aplikasi pendeteksi plagiarisme kode program dalam bahasa C.
3. Meneliti performa dari algoritma *Levenshtein Distance* dan *Brute Force* dalam mengidentifikasi bobot persentase kecenderungan plagiarisme antara dokumen kode program dalam bahasa C
4. Mengimplementasikan teknik *preprocessing* agar dapat meningkatkan bobot persentase kecenderungan plagiarisme yang dihasilkan oleh Algoritma *Levenshtein Distance* dan *Brute Force*.

1.5 Manfaat Penelitian

Penelitian ini secara khusus bermanfaat bagi tenaga pengajar (dosen) mata kuliah pemrograman menggunakan bahasa C dalam melakukan pemeriksaan terhadap kecenderungan plagiarisme antara kode program yang satu dengan yang lainnya,

dengan usaha yang lebih kecil, ketelitian yang lebih tinggi, serta waktu yang lebih cepat.

Selain itu, secara umum penelitian ini juga bermanfaat sebagai dasar bagi para peneliti lain yang tertarik untuk melanjutkan penelitian dengan topik yang serupa.

1.6 Metode Penelitian

Metode yang digunakan dalam penelitian ini terdiri dari langkah-langkah sebagai berikut.

1. Studi Literatur

Melakukan studi kepustakaan melalui hasil penelitian orang lain maupun artikel-artikel atau paper-paper lainnya yang relevan, serta mempelajari teori-teori tentang plagiarisme kode program dan teknik yang tepat untuk diimplementasikan dalam aplikasi pendeteksi plagiarisme kode program dalam bahasa C.

2. Observasi

Melakukan wawancara dan pengamatan terhadap masalah plagiarisme kode program dalam ruang lingkup mahasiswa fakultas ICT (*Information and Communication Technology*) di Universitas Multimedia Nusantara untuk mata kuliah Algoritma dan Pemrograman.

3. Analisis dan Perancangan Sistem

Melakukan analisis terhadap masalah plagiarisme kode program yang ditemukan serta menentukan batasan-batasan dan kebutuhan yang diperlukan dalam membuat aplikasi pendeteksi plagiarisme kode program dalam bahasa C.

4. Implementasi Algoritma

Mengimplementasikan algoritma yang telah dirancang sebelumnya untuk membangun aplikasi tersebut.

5. Pengujian

Melakukan pengujian perbandingan antara beberapa kode program dalam bahasa C dengan menggunakan aplikasi yang telah dibuat. Kemudian kode program dalam bahasa C tersebut juga dibandingkan secara manual, lalu dilakukan analisis terhadap hasil dan performa aplikasi maupun algoritmanya.

1.7 Sistematika Penulisan Laporan Penelitian

Dalam penulisan skripsi ini, sistematika penulisan dibagi menjadi lima bab, yaitu.

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode, serta sistematika penulisan laporan penelitian.

2. BAB II TELAAH LITERATUR

Bab ini berisi teori-teori yang digunakan dalam perancangan, implementasi, dan analisis penelitian, yang terdiri dari teori-teori tentang rekayasa perangkat lunak, plagiarisme, bahasa pemrograman C, serta penjelasan tentang teknik *preprocessing* hingga algoritma *Levensthein Distance* dan *Brute Force* yang digunakan dalam sistem pendeteksi plagiarisme.

3. BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi gambaran umum tentang sistem yang ada sekarang sesuai studi kasus yang digunakan, kekurangan dan masalah yang dihadapi dari sistem yang sekarang serta usulan pemecahan atas masalah yang dihadapi. Selain itu juga dijelaskan spesifikasi, tujuan, batasan, masukan dan keluaran sistem, serta gambaran perancangan sistem, mulai dari proses yang terjadi di dalamnya, desain *interface*, hingga struktur hirarki menu.

4. BAB IV IMPLEMENTASI DAN EVALUASI

Pada bab ini, dipaparkan hasil penelitian, mulai dari proses implementasi dari sistem yang dibuat, spesifikasi perangkat lunak maupun perangkat keras yang digunakan dalam pembangunan sistem, proses pengujian sistem, hasil pengujian sistem, penjelasan cara pemakaian sistem serta evaluasi akhir dari sistem yang dibuat.

5. BAB V SIMPULAN DAN SARAN

Bagian terakhir laporan penelitian ini berisi simpulan dan saran. Pada simpulan, diuraikan mengenai jawaban atas rumusan masalah yang diuraikan pada BAB I, beserta informasi tambahan yang diperoleh atas dasar temuan penelitian. Sedangkan pada bagian saran, diuraikan manifestasi dari penulis atas sesuatu yang belum ditempuh dan layak untuk dilaksanakan pada penelitian lanjutan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA