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.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini proses pertukaran informasi menjadi sangat mudah. Kemajuan yang cukup besar di bidang komputer dan dunia internet semakin mempercepat proses tersebut. Perkembangan teknologi ini memang membawa dampak yang positif tetapi ada pula hal yang negatif yang tidak dapat dihindari yaitu tindakan plagiarisme. Praktik plagiat merupakan hal yang tidak asing lagi dan seringkali terjadi diberbagai kalangan. Dalam Kamus Besar Bahasa Indonesia, definisi plagiarisme atau sering juga disebut plagiat adalah pengambilan atau penjiplakan karya ataupun pendapat orang lain dan menjadikannya seolah-olah karya atau pendapat sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri.

Plagiarisme adalah suatu kejahatan akademik dan merupakan pelanggaran etika. Dengan demikian, plagiarisme pada dasarnya adalah suatu bentuk tindakan korupsi etika. Bagi kalangan akademisi, plagiarisme adalah aib yang sangat tidak mudah terhapuskan (FX Sugiyanto, 2010).

Menurut data pada situs *Plagiarism.org*, terdapat sebuah penelitian yang dilakukan oleh *The Center for Academic Integrity* dengan hasil bahwa hampir 80% mahasiswa mengaku pernah melakukan penjiplakan sekurang-kurangnya satu kali. Demikian juga dari hasil survei yang dilakukan oleh *Psychological*

Record, 36% mahasiswa mengaku pernah melakukan penjiplakan terhadap dokumen tertulis.

Dalam dunia perkuliahan khususnya pada jurusan-jurusan yang mendapatkan mata kuliah pemprograman tindakan plagiarisme mudah terjadi di kalangan mahasiswa pada saat pengerjaan tugas pembuatan kode program dengan berbagai alasan. Cara yang umum dan mudah dilakukan untuk melakukan plagiat kode program adalah dengan melakukan *Naked Plagiarism* yang secara garis besar berarti melakukan teknik *copy-paste,* ataupun dengan melakukan sedikit perubahan setelah melakukan *copy-paste* yang dikenal dengan *Semi Naked Plagiarism* (Sendow, 2011).

Begitu mudahnya melakukan plagiarisme di kalangan mahasiswa memunculkan ide penulis untuk mengembangkan aplikasi pendeteksi plagiarisme untuk kode program berbahasa C. Berbeda dengan aplikasi pendeteksi plagiarisme untuk dokumen teks dalam bentuk uraian, aplikasi pendeteksi plagiarisme kode program harus dibuat lebih spesifik, sesuai dengan karakteristik dari kode program tersebut. Untuk itu dibutuhkan algoritma yang dapat mendeteksi kecenderungan plagiarisme kode program dengan baik.

Adapun penelitian sebelumnya yang dilakukan oleh Andreas Arifianto (Arifianto, 2011) dengan judul "Rancang Bangun Aplikasi Pendeteksi Plagiarisme Kode Program Dalam Bahasa C Menggunakan Algoritma *Levenshtein Distance* dan *Brute Force*" telah menjadi acuan bagi penulis dalam melakukan penelitian. Dalam aplikasi tersebut ditemukan masih diperlukan pengembangan dari batasan yang telah ada sebelumnya dalam mendeteksi kode program, serta berdasarkan

penelitian yang pernah dilakukan mengenai perbandingan kecepatan 4 buah algoritma *string matching Brute Force (BF), Knuth Morris Pratt (KMP), Boyer Moore (BM)* dan *Rabin Karb (RK)*, didapatkan kesimpulan pada masing-masing algoritma adalah sebagai berikut : BM : 0,92 detik , BF : 0,98 detik, KMP : 0,99 detik dan RK : 3,46 detik dengan perulangan sebanyak 100 kali, serta ditinjau dari jumlah perbandingan pola dengan teks, secara signifikan *Boyer Moore* menjadi algoritma yang lebih unggul. (Darmawan Utomo, Eric W. Harjo, Handoko, 2008).

Algoritma *Boyer Moore* lebih efektif dibandingkan algoritma lainnya dan telah dibuktikan oleh Richard Cole dengan membandingkan kompleksitas algoritma lain (Redya Febriyanto, Sherry Bayu, Hanindyo Wirotomo, 2006). Penelitian-penelitian tersebutlah yang melatar belakangi penulis untuk melakukan penelitian dengan judul "Peningkatan Kemampuan Aplikasi Pendeteksi Plagiarisme Pada Kode Program Berbahasa Pemprograman C Dengan Algoritma *Boyer Moore* Dan *Brute Force"*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diuraikan rumusan masalahnya yaitu

1. Bagaimana membuat aplikasi untuk mendeteksi plagiarisme pada kode program mahasiswa?
2. Bagaimana meningkatkan kemampuan aplikasi pendeteksi plagiarisme yang menggunakan algoritma *Levenshtein Distance* dengan algoritma *Boyer Moore*?

UNIVERSITAS MULTIMEDIA NUSANTARA

1.3 Batasan Masalah

Batasan masalah dalam skripsi ini:

1. Hanya mendeteksi dokumen kode program berbahasa pemprograman C.
2. Data yang diuji berformat .c atau .cpp.
3. Materi kode program yang dibandingkan antara lain *variable, conditional, looping, function, array, pointer, struct* dan *file.*

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah

1. Membangun aplikasi pendeteksi plagiarisme pada kode program bahasa C
2. Meningkatkan kemampuan pendeteksi plagiarisme yang telah dibuat terdahulu dengan memodifikasi teknik *preprocessing* dan menggunakan algoritma *Boyer Moore* dan *Brute Force*.

1.5 Manfaat Penelitian

Penelitian ini bermanfaat untuk membantu proses pemeriksaan praktik plagiarisme yang terjadi di kalangan akademis secara khusus di kalangan mahasiswa. Hasil dari penelitian ini diharapkan dapat membantu para pengajar mata kuliah yang menggunakan bahasa pemprograman C untuk dapat mengetahui tindak plagiarime yang terjadi dengan memberikan nilai persentase kecenderungan plagiarisme berdasarkan kemiripan yang terjadi.

1.6 Metode Penelitian

Metode yang digunakan dalam penelitian ini terdiri dari:

1. Studi Literatur

   Melakukan studi kepustakaan dari hasil penelitian yang telah dilakukan sebelumnya oleh orang lain dan artikel–artikel yang terkait dengan pembuatan pendeteksi plagiarisme, serta mempelajari teknik dan algoritma yang tepat untuk dapat meningkatkan kemampuan aplikasi pendeteksi plagiarisme yang sudah dilakukan oleh peneliti sebelumnya.

2. Analisis Data

   Melakukan analisis untuk menentukan batasan–batasan masalah mengenai aplikasi pendeteksi plagiarisme dan kebutuhan-kebutuhan yang diperlukan dalam pembuatan dan peningkatan kemampuan aplikasi pendeteksi plagiarisme yang dari penelitian sebelumnya.

3. Perancangan

   Melakukan perancangan algoritma yang paling tepat dan teknik-teknik yang diperlukan untuk digunakan dalam membangun dan meningkatkan aplikasi pendeteksi plagiarism.

4. Implementasi

   Melakukan implementasi algoritma dan teknik yang telah ditentukan pada metode sebelumnya untuk membangun dan maningkatkan aplikasi tersebut.

5. Pengujian

UNIVERSITAS MULTIMEDIA NUSANTARA

Melakukan pengujian pada aplikasi yang telah dibangun dengan membandingkan kode program berbahasa C dan menganalisis hasil dan performa aplikasi yang sudah dibangun dengan performa aplikasi yang sudah dibangun sebelumnya.

6. Penulisan Laporan

   Melakukan penulisan laporan dari penelitian yang telah dilakukan mulai dari tahap studi literatur hingga pengujian pada aplikasi tersebut.

1.7 Sistematika Penulisan Laporan Penelitian

Dalam penulisan skripsi ini, sistematika penulisan dibagi menjadi lima (5) 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, yang terdiri dari teori-teori tentang rekayasa perangkat lunak, plagiarisme, *Boyer Moore* dan *Brute Force* yang digunakan dalam sistem pendeteksi plagiarisme.

3. BAB III ANALISIS DAN PERANCANGAN

   Bab ini berisi gambaran umum tentang sistem yang ada sekarang sesuai studi kasus yang digunakan, kekurangan dan masalah yang dihadapi dari

sistem yang dibuat sebelunya serta usulan pemecahan atas masalah yang dihadapi. Dijelaskan juga mengenai spesifikasi, tujuan, batasan, masukkan 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 PENGUJIAN

   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 batasan masalah serta tujuan penelitian yang diuraikan pada BAB I, beserta informasi tambahan yang diperoleh atas dasar temuan penelitian. Pada bagian saran, dijelaskan saran dari penulis mengenai pengembangan selanjutnya yang dapat dilakukan oleh peneliti selanjutnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA