

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi dan komunikasi tidak hanya membawa dampak positif, tetapi juga membawa dampak negatif, salah satu bentuk tindakan negatif tersebut adalah menyontek. Era globalisasi ditandai dengan perkembangan ilmu pengetahuan dan teknologi yang sangat pesat. Pengelolaan informasi dengan menggunakan komputer yang dilatarbelakangi perbuatan curang dan rendahnya kemampuan mahasiswa dalam berkreasi dan berinovasi memicu tindakan negatif. Salah satunya adalah tindakan plagiarisme (Kharisman, 2012). Plagiarisme sering dikaitkan dengan perilaku menyontek. Plagiarisme bisa menjadi masalah besar di universitas-universitas. Di tahun 2009 China Association for Science and Technology (CAST) melaporkan 43,4% dari kira-kira 30.000 peneliti mendeskripsikan plagiarisme adalah masalah yang sangat serius (Heitman, 2010). Lebih dari setengahnya mengatakan perbuatan tersebut berada di sekitar mereka.

Plagiarisme adalah tindakan produksi ulang, penggunaan karya orang lain tanpa pemberian dan penyaduran sumber (Juričić, 2011), (Kustanto & Liem, 2009). Plagiarisme dalam Kamus Besar Bahasa Indonesia (KBBI) adalah penjiplakan atau pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri (Departemen Pendidikan Nasional, 2014). Menurut pasal 1 Peraturan Menteri, plagiat adalah perbuatan secara sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai (Peraturan Menteri Pendidikan Nasional Republik Indonesia, 2010).

Berdasarkan penelitian (Gipp & Norman, 2011) mengategorikan praktik plagiat berdasarkan cara yang digunakan, diantaranya

1. *Copy & Paste plagiarism*, menyalin setiap kata tanpa perubahan
2. *Disguised plagiarism*, tergolong kedalam praktik menutupi bagian yang disalin, teridentifikasi ke dalam empat teknik, yaitu *shake & paste*, *expensive plagiarism*, *contractive plagiarism*, dan *mosaic plagiarism*.

3. *Technicaldisguise*, teknik meringkas untuk menyembunyikan konten plagiat dari deteksi otomatis dengan memanfaatkan kelemahan dari metode analisis teks dasar, misal dengan mengganti huruf dengan simbol huruf asing.
4. *Undueparaphrasing*, sengaja menuliskan ulang pemikiran asing dengan pemilihan kata dan gaya plagiator dengan menyembunyikan sumber asli.
5. *Translatedplagiarism*, mengkonversi konten dari satu bahasa ke bahasa lain.
6. *Ideaplaciarism*, menggunakan ide asing tanpa menyatakan sumber.
7. *Selfplagiarism*, penggunaan sebagian atau keseluruhan tulisan pribadi yang tidak dibenarkan secara ilmiah.

Telah banyak upaya yang ditujukan untuk mendeteksi dan mencegah plagiarisme, salah satunya melalui peraturan dan pemberian sanksi yang berat. Salah satunya melalui Peraturan Menteri Pendidikan Nasional Nomor 17 Tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi. Ancaman berat bagi seorang civitas akademika yang terbukti melakukan plagiarisme, mulai dari teguran sampai diberhentikannya pendidikan seorang mahasiswa, dan pencabutan gelar akademik bagi seorang yang terbukti melakukan tindakan plagiarisme (Peraturan Menteri Pendidikan Nasional Republik Indonesia, 2010).

Plagiarisme kode program merupakan masalah besar di pendidikan jurusan komputer (Cosma & Joy, 2008). Meskipun plagiarisme dapat dilakukan dengan mudah, mendeteksi perilaku ilegal ini bukan tugas yang sepele. Plagiarisme bisa dilakukan oleh setiap mahasiswa, bahkan yang paling cerdas dan mengerti konsep pemrograman dengan baik dapat melakukan tindakan tersebut (Kavcic & D., 2009), (Karnalim, 2016). Plagiarisme yang dilakukan siswa mungkin akan bervariasi, seperti yang sederhana dengan menyalin sama persis (dilakukan oleh mahasiswa amatir) atau yang paling rumit yang menggabungkan berbagai logika dan perubahan struktur (dilakukan oleh siswa cerdas) (Rabbani & Karnalim,

2016). Parker dan Hamblen menganggap program yang dapat diproduksi dari program lain dengan sejumlah transformasi kecil dianggap pelagiat bahkan jika tidak persis mengandung unsur-unsur yang sama dari program atau kode orang lain (Hamblen & Parker, 1989). Transformasi bisa saja sederhana, seperti merubah nama variabel, memodifikasi komentar, menambah fungsi baru dan lain-lain.

Sejumlah penelitian telah dilakukan dalam membangun perangkat lunak yang dapat mendeteksi karya cipta seseorang, dari aplikasi tersebut dapat diketahui tingkat kemiripan antar dokumen, diantaranya *Lavenshtein Distance Calculatioin Using Dynamic Programming for Source Code Plagiarism Checking* oleh Tito D. Kesumo Siregar (2014), *Detection Source Code Similarity using Low-Level Language* (Juričić, 2011) dan YAP3 yang menggunakan Rabin-Karp Greedy-String-Tiling (Wise, 1996)

Pada penelitian Tito D. Kesumo Siregar (2014) dilakukan penelitian bagaimana menerapkan algoritma *levenshteindistance* dalam mendeteksi kemiripan isi sebuah kode program. *Levenshtein distance* merupakan metrik yang digunakan untuk mengukur keterbedaan jarak antara dua sekuens. *Levenshtein distance* antara dua string ditentukan berdasarkan jumlah minimum perubahan/pengeditan yang diperlukan untuk melakukan transformasi dari satu bentuk string ke bentuk string lain. Hasil yang diperoleh dari penelitian tersebut dapat digunakan untuk mendeteksi plagiarisme dengan hasil yang cukup baik, nilai terendah untuk menentukan apakah kode program merupakan plagiarisme sekitar 70%.

Vedran Juričić dalam Penelitiannya *Detection Source Code Similarity using Low-level Language*(Juričić, 2011) mendeteksi kemiripan kode program .NET yang sama persis dengan membandingkan *low-levelintermediatelanguage*. Metode ini menunjukkan hasil yang kurang baik ketika terdapat perubahan kode program, penataan ulang statement kode program, dan perubahan posisi member dari *class*.

Oscar Karnalim dan Faqih Salban Rabbani dalam penelitiannya yang berjudul *Detecting Source Code Plagiarism on .NET Programming Language using Adaptive Local Alignment* (Rabbani & Karnalim, 2016) dengan deteksi plagiarisme berfokus pada bahasa pemrograman .NET. kode program .NET dikonversikan ke .NET Common Intermediate Language (CIL) dan similaritasnya ditentukan berdasarkan algoritma Adaptive Local Alignment (ALA). Dalam hal efisiensi, CIL lebih efisien dibandingkan STDS (Standard token-based approach) yang mengurangi berbagai proses yang disebabkan karena ukuran token yang terbatas. Penelitian ini mengurangi sekitar 80,691% proses ketika dibandingkan dengan STDS berdasarkan dataset milik Karnalim (Karnalim, 2016)

Komparasi struktural menjadi diterima secara luas untuk komparasi kode program. Beberapa metode komparasi struktural yang penting antara lain greedy-string-tiling (Prechelt, Malpohl, & Philippsen, 2002), *localalignment* (Gitchell & Tran, 1999), dan *parsetreecomparison* (Son, Park, & Park, 2006). Penentuan kemiripan suatu dokumen dilakukan berdasarkan kesamaan struktural dokumen. Algoritma Smith-Waterman memiliki keakuratan yang lebih baik pada saat membandingkan dokumen yang mengandung perubahan struktur (Thalib & Kusumawati, 2014).

Sebagian besar sistem pendeteksi tersebut membandingkan kode program komputer secara sintaksis. Tetapi tidak semua potongan kode program dengan makna yang sama memiliki bentuk sintaks yang serupa. Penulisan pada kode program memiliki bentuk yang berupa-rupa tergantung penulis kode program, seperti baris instruksi *for* dan *while*, penulisan komen dan variabel yang berbeda-beda. Secara sintaks berbeda, akan tetapi secara makna mirip.

Kebanyakan bahasa yang paling banyak dan pertama kali digunakan dalam pelajaran bahasa pemrograman di universitas adalah C (Kim, 2015). Bahasa C menjadi bahasa pemrograman yang populer di Indonesia karena hampir sebagian besar perguruan tinggi jurusan IT ataupun lembaga kursus pemrograman menggunakan bahasa ini untuk dipelajari pertama kali sebelum mempelajari bahasa pemrograman lain (Priyanto, 2014). Di universitas terutama jurusan yang

mempelajari bahasa pemrograman menggunakan bahasa C sebagai media untuk memahami konsep pemrograman. Contohnya MIT yang mengadakan kelas “*Practical Programmin in C*”.

Berdasarkan permasalahan tersebut maka akan dibangun sebuah sistem pendeteksi yang membandingkan kode program bahasa C yang sebelumnya dilakukan *pre-processing* dan dibuat token untuk setiap karakter/string yang berada dalam satu *keyword*. Token inilah yang akan dibandingkan oleh algoritma Smith-Waterman. Harapannya pendeteksian kemiripan pada kode program yang terdapat perubahan struktur dapat juga terdeteksi dengan baik. Semoga pembuatan sistem ini akan membantu deteksi dan meminimalisir tindak plagiarisme di lingkungan pendidikan komputer.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang dikemukakan, maka penyusun memiliki rumusan masalah.

1. Bagaimana membangun sistem pendeteksi kemiripan pada kode program bahasa C dengan menggunakan algoritma Smith-Waterman?
2. Bagaimana membandingkan kemiripan dokumen program bahasa C secara sintaksis yang terdapat perbedaan struktur dan penulisan?

1.3 Tujuan Penelitian

Bedasarkan pada rumusan masalah yang dipaparkan, maka penyusun memiliki tujuan penelitian sebagai berikut.

1. Membuat sistem untuk membantu mendeteksi kemiripan pada dokumen kode program bahasa C.
2. Melakukan implementasi Algoritma Smith-Waterman untuk membuat sistem pendeteksi kemiripan pada dokumen kode program bahasa C.

1.4 Manfaat Penelitian

Manfaat bagi Pengguna Aplikasi

1. Memberikan kemudahan dalam pendeteksian kemiripan di dokumen program bahasa C yang memiliki kemiripan dalam penulisan.
2. Membantu instansi pendidikan dalam mendeteksi tindak mencontek kode program sehingga bisa dipakai sebagai bahan pertimbangan untuk mendeteksi adanya tindak plagiarisme.

Manfaat bagi Peneliti

1. Penelitian ini memberikan pengalaman dan pengetahuan mengenai pembuatan sistem pendeteksi kemiripan dokumen kode program.

1.5 Batasan Masalah

Adapun parameter batasan masalah yang dilakukan dalam penelitian ini, yaitu:

1. Kode program yang dideteksi merupakan 2 kode program bahasa C.
2. Hasil perbandingan dengan sistem yang dibuat tidak menunjukkan bahwa mutlak telah atau tidak terjadi tindak plagiarisme antar dua dokumen program. Hasil akhir sistem hanya memberikan nilai similaritas antar dokumen.
3. Plagiarisme yang dibahas di penelitian ini adalah ruang lingkup masalah. Sistem ini memberikan nilai kemiripan, nilai kemiripan itu yang akan menjadi acuan keputusan tindak plagiarisme.

1.6 Sistematika Penulisan

Berikut merupakan sistematika penulisan karya tulis yang berjudul “Sistem Pendeteksi Kemiripan pada Kode Program Bahasa C Menggunakan Algoritma Smith-Waterman” sebagai berikut:

BAB I PENDAHULUAN

Di dalam Bab ini akan dijelaskan latar belakang masalah yang menjadi alasan penulis mengangkat topik ini sebagai skripsi. Dimulai dari kegelisahan peneliti

Rendy Reynaldy Anggradwiguna, 2017

SISTEM PENDETEKSI KEMIRIPAN PADA DOKUMEN KODE PROGRAM BAHASA C MENGGUNAKAN ALGORITMA SMITH-WATERMAN

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

terhadap tindak mencontek dan menjiplak hasil karya orang lain, khususnya dalam pendidikan komputer. Karena dengan tindakan tersebut menyebabkan anak didik yang sedang belajar konsep pemrograman melakukan jalan pintas demi dapat melewati ujian dan tugas. Menjadikan peneliti ingin membuat sistem pendeteksi kemiripan dokumen kode program. Selain latar belakang terdapat juga rumusan masalah sehingga penelitian ini terfokus pada satu masalah yang dikaji dan tidak melebar ke permasalahan yang lain, rumusan masalah yang pertama adalah mengenai bagaimana membuat sistem pendeteksi kemiripan dokumen kode program bahasa C dengan menggunakan algoritma Smith-Waterman, kemudian rumusan yang kedua adalah bagaimana membandingkan dokumen kode program bahasa C secara sintaksis yang terdapat perubahan struktur dan penulisan. Kemudian setelah menjelaskan latar belakang dan rumusan masalah, terdapat tujuan penelitian, manfaat penelitian, dan batasan masalah yang disampaikan peneliti dalam penelitian ini. Pada bagian akhir dijelaskan sistematika penulisan yang akan menjadi pedoman dalam penulisan skripsi, strukturnya adalah Bab I Pendahuluan, Bab II Kajian Pustaka, Bab III Metode Penelitian, dan Bab IV Hasil dan Pembahasan.

BAB II KAJIAN PUSTAKA

Kajian pustaka menjadi bagian yang penting dalam penelitian ini. Di dalam kajian pustaka, peneliti membahas mengenai teori-teori pendukung yang berkaitan dengan Sistem pendeteksi dokumen kode program. Mulai dari konsep *textmining* dan *preprocessing*, bahasa C yang akan dilakukan pendeteksian, sampai algoritma Smith-Waterman. Teori yang didapatkan bersumber dari buku-buku, jurnal internasional dan *electronic book* (e-book) yang digunakan sebagai referensi oleh peneliti yang berkaitan dengan penelitian.

BAB III METODOLOGI PENELITIAN

Metode penelitian membahas mengenai metode dan teknik penelitian, termasuk bagaimana peneliti mencari berbagai sumber yang sesuai dengan topik penelitian

hingga mengolah sumber-sumber yang diperoleh. Metode penelitian yang dipakai pada penelitian kali ini dimulai dari perumusan masalah, kemudian studi literatur, desain dan perancangan dan akhirnya pada tahap dokumentasi. Selain metode penelitian, ada pula model pengembangan sistem yang digunakan dalam mengembangkan sistem. Model yang digunakan adalah model sekuensial linier. Model sekuensial linier dimulai dari analisis, desain, kode, dan tes.

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini menjelaskan mengenai seluruh isi penelitian yang merupakan pertanyaan dari rumusan masalah yang telah disetujui oleh dosen pembimbing. Pemaparan dalam bab ini diuraikan secara deskriptif untuk menjawab pertanyaan-pertanyaan secara terperinci. Untuk lebih memudahkan pemahaman, dalam bab ini peneliti membagi bab kedalam beberapa sub bab judul. Pertama, membahas analisis, analisis kebutuhan sistem, batasan sistem dan analisis umum. Kedua, mengenai desain perancangan sistem yang dimulai dari perancangan antarmuka, algoritma dan *flowchart* sistem. Ketiga, Implementasi dari hasil perancangan yang sudah dibuat pada tahap desain. Keempat, pengujian sistem yang telah dibuat dan dipaparkan pula hasil pengujian sistem tersebut.

BAB V KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan atas pembahasan yang sudah dikaji oleh peneliti melalui penafsiran secara umum yang diperoleh berdasarkan hasil penelitian mengenai sistem pendeteksi kemiripan dokumen kode program bahasa C dengan menggunakan algoritma Smith-Waterman, selain kesimpulan, bab ini juga memberikan saran serta masukan bagi peneliti yang akan melakukan penelitian atau meneruskan penelitian yang sejenis, saran di sini berisi kelemahan dan kekurangan yang dialami oleh peneliti.