



### **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 saat ini melaju semakin pesat. Dengan adanya internet dan media komunikasi lain, pertukaran informasi pun dapat dilakukan dengan cepat dan bebas. Selain keuntungan, kecepatan dan kebebasan dalam pertukaran informasi ini juga membawa dampak buruk. Informasi dan opini yang bersifat publik dapat diakses dimanapun oleh siapapun. Hal ini memacu terjadinya plagiarisme pada berbagai kalangan untuk berbagai tujuan.

Dalam Kamus Besar Bahasa Indonesia, plagiarisme didefinisikan sebagai pengambilan atau penjiplakan karya ataupun pendapat orang lain dan menjadikannya seolah-olah karya atau pendapat sendiri. Plagiarisme sendiri merupakan kejahatan akademik dan pelanggaran etika. Di Indonesia, pencegahan dan penanggulangan plagiarisme pada Perguruan Tinggi sudah diatur oleh Permendiknas Nomor 17 Tahun 2010 mengenai Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi.

Dalam dunia akademik, plagiarisme bukanlah kasus yang jarang ditemui. Pada jurusan-jurusan yang mendapatkan mata kuliah pemrograman, plagiarisme umum terjadi dalam pengerjaan tugas. Berdasarkan penelitian yang diadakan oleh Nadelson seperti yang tercantum pada laporan penelitian oleh Cosma pada tahun 2008, dari 72 orang yang diwawancarai diduga terdapat 570 kasus plagiarisme yang dilakukan oleh mahasiswa sarjana maupun pasca sarjana. Dari sejumlah kasus tersebut, terdapat 134 kasus plagiarisme tidak disengaja oleh mahasiswa sarjana dan 39 kasus oleh mahasiswa pasca sarjana.

Mendeteksi kasus plagiarisme bukanlah hal yang mudah. Berdasarkan hasil wawancara yang dilakukan oleh Arfianto pada tahun 2011, terdapat beberapa kendala-kendala yang harus dihadapi dalam melakukan pendeteksian plagiarisme pada kode program untuk tugas akademik. Kendala-kendala tersebut adalah banyaknya dokumen kode program yang harus diperiksa dalam waktu terbatas, ketelitian dalam membandingkan kode program, konsep kelas paralel, dan dokumentasi hasil pendeteksian dokumen kode program dalam jumlah yang cukup banyak. Untuk mempermudah proses pendeteksian tersebut, diperlukan sebuah aplikasi untuk menghitung kecenderungan plagiarisme pada dua dokumen kode.

Penelitian mengenai aplikasi pendeteksi plagiarisme sudah dilakukan oleh banyak kalangan. Diantaranya adalah penelitian yang dilakukan oleh Andreas Arfianto dengan judul “Rancang Bangun Aplikasi Pendeteksi Plagiarisme Kode Program Dalam Bahasa C Menggunakan Algoritma *Levenshtein Distance* dan *Brute Force*” yang kemudian ditingkatkan oleh Regina Natalia (2011) dalam penelitiannya yang berjudul “Peningkatan Kemampuan Aplikasi Pendeteksi Plagiarisme pada Kode Program Berbahasa Pemrograman C dengan Algoritma *Boyer Moore* dan *Brute Force*”. Disamping itu, terdapat juga penelitian yang dilakukan oleh Ahmad Gull Liaqat dan Aijaz Ahmad (2011) yang berjudul “*Plagiarism Detection in Java Code*”. Ketiga penelitian ini digunakan oleh penulis sebagai acuan dalam melakukan penelitian.

Aplikasi milik Ahmad Gull Liaqat dan Aijaz Ahmad menggunakan algoritma pembandingan yang sama dengan aplikasi yang dibangun oleh Andreas Arfianto, yaitu *Levenshtein Distance*. Aplikasi milik Andreas Arfianto sendiri memiliki

kelemahan pada algoritma pembandingnya, dimana aplikasi tersebut masih dapat tertipu dengan perubahan pada struktur kode. Kelemahan ini kemudian diperbaiki pada aplikasi milik Regina Natalia dengan melakukan modifikasi pada teknik *preprocessing* yang digunakan, dan mengganti algoritma *Levenstein Distance* dengan algoritma *Boyer Moore*. Atas pertimbangan tersebut penulis mempertimbangkan algoritma *Boyer Moore* sebagai algoritma dasar dari penelitian. Algoritma *Brute Force* tidak digunakan karena selain memakan waktu dan tidak efisien, berdasarkan data penelitian milik Regina Natalia, algoritma tersebut memiliki persentase ketepatan yang lebih kecil dibandingkan dengan algoritma *Boyer Moore*.

Lebih lanjut lagi, Domenico Cantoned dan Simone Faro (2003) dalam jurnalnya yang berjudul "*Forward Fast Search: Another Fast Variant of the Boyer-Moore String Matching Algorithm*" memodifikasi algoritma *Boyer Moore* untuk meningkatkan efisiensi waktu dari algoritma tersebut dan menghasilkan variasi baru dari algoritma *Boyer Moore* yang kemudian disebut sebagai algoritma *Forward Fast Search*. Dalam penelitian tersebut, algoritma *Forward Fast Search* dibandingkan dengan algoritma *Horspool*, *Tuned Boyer-Moore*, *Quick Search*, *Berry-Ravidran*, dan *Fast Search*. Berdasarkan hasil penelitian yang sama, algoritma ini memiliki waktu *run time* paling kecil pada sebagian besar kasus, meski masih dikalahkan oleh algoritma *Berry-Ravidran* dan *Fast Search* dalam beberapa kasus, terutama dalam membandingkan teks dengan *Natural Language*. Atas dasar penelitian ini, penulis memilih untuk menggunakan algoritma *Forward Fast Search* sebagai ganti algoritma *Boyer Moore* dalam melakukan penelitian.

Disamping algoritma, penulis memutuskan untuk menggunakan *website* sebagai basis dari aplikasi yang dibangun. Hal ini dilakukan atas pertimbangan bahwa aplikasi berbasis *website* dan *mobile* lebih mudah diakses dibandingkan dengan aplikasi berbasis *desktop*. Lebih lanjut lagi, aplikasi *mobile* kurang memadai untuk melakukan pendeteksian plagiarisme pada kode program, karena pada umumnya kode program tidak ditulis pada *gadget* yang bersifat *mobile* dan aplikasi *mobile* memiliki *resource* yang lebih terbatas dibandingkan dengan aplikasi *website*.

Berdasarkan latar belakang tersebut, penulis melakukan penelitian dengan judul “Rancang Bangun Aplikasi Pendeteksi Plagiarisme pada Kode Program Java Menggunakan Algoritma *Forward Fast Search*”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat dirumuskan bahwa masalah yang dibahas adalah sebagai berikut.

1. Bagaimana cara membangun aplikasi pendeteksi plagiarisme untuk bahasa pemrograman Java berbasis web dengan menggunakan PHP?
2. Bagaimana cara mengimplementasikan algoritma *Forward Fast Search* dalam mendeteksi plagiarisme pada kode program berbahasa Java?

## 1.3 Batasan Masalah

Masalah-masalah yang dibatasi dalam penelitian ini adalah

1. Hanya mendeteksi dokumen kode program berbahasa pemrograman Java.
2. Diutamakan untuk pendeteksian plagiarisme pada kode program yang bersifat akademis, bukan komersial.

3. Hanya menunjukkan persentase kesamaan kode dan tidak memberikan keputusan mutlak, sehingga masih dibutuhkan pengecekan manual terhadap dokumen-dokumen dengan persentase kesamaan tinggi.
4. Elemen kode program yang dibandingkan antara lain *class*, *method*, variabel (numerik, karakter, *boolean*, dan *object*), *array*, *looping*, *conditional statement*, *conditional ternary*, operasi matematika (*prefix*, *infix*, dan *postfix*), dan *compound operator*. Tidak memperhitungkan elemen *Graphical User Interface* (GUI) pada kode.
5. Jenis *file* arsip yang digunakan untuk mengupload kode hanya meliputi *file ZIP* dan *TAR*.
6. Diasumsikan kode yang akan diperiksa tidak memiliki kesalahan *syntax*.

#### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah

1. Membangun aplikasi berbasis web untuk menghitung persentase kesamaan pada kode program Java sederhana terutama untuk bidang akademis.
2. Menerapkan algoritma *Forward Fast Search* dalam menghitung persentase kesamaan antara kode.

#### 1.5 Manfaat Penelitian

Penelitian ini bermanfaat untuk membantu proses pendeteksian plagiarisme pada kode program berbahasa Java, khususnya pada bidang akademis. Hasil dari penelitian ini dapat membantu para pengajar untuk menyelidiki kasus plagiarisme pada mata kuliah pemrograman yang menggunakan bahasa Java.

Selain itu, penelitian ini juga dapat digunakan sebagai landasan untuk peneliti lain yang ingin meneliti topik yang serupa.

### 1.6 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 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan dalam perancangan, implementasi, dan analisis penelitian yang terdiri dari teori-teori tentang plagiarisme, algoritma *Boyer Moore*, algoritma *Horspool*, algoritma *Fast Search*, algoritma *Forward Fast Search*, dan *preprocessing* yang digunakan dalam sistem pendeteksi plagiarisme.

#### 3. BAB III METODOLOGI DAN PERANCANGAN

Bab ini berisi metode penelitian yang digunakan, gambaran umum tentang sistem yang ada sekarang sesuai studi kasus yang digunakan, kekurangan dan masalah yang dihadapi dari sistem yang sudah ada serta usulan pemecahan atas masalah yang dihadapi. Selain itu, pada bab ini dijelaskan juga spesifikasi, tujuan, batasan, masukan dan keluaran sistem, dan gambaran sistem, mulai dari proses yang terjadi, desain *interface*.

#### 4. BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini, diaparkan 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 dari laporan penelitian ini berisi simpulan dan saran. Pada simpulan, diuraikan jawaban atas rumusan dan tujuan masalah yang diuraikan pada Bab I, beserta informasi tambahan yang diperoleh atas dasar temuan penelitian. Pada bagian saran, dijelaskan saran dari penulis mengenai pengembangan yang dapat dilakukan oleh peneliti selanjutnya.

# UMMN