

BAB I

PENDAHULUAN

1.1 Latar Belakang

Plagiarisme atau sering juga disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri [1]. Plagiat dapat dianggap sebagai tindak pidana karena mencuri hak cipta orang lain. Di dunia pendidikan, pelaku plagiarisme dapat mendapat hukuman berat seperti dikeluarkan dari sekolah atau universitas. Pelaku plagiat disebut sebagai plagiator.

Di dunia pendidikan, khususnya universitas, rentan terjadi plagiarisme. Misalnya dalam pembuatan skripsi, sering terjadi kemiripan antar skripsi. Plagiarisme dapat mematikan kreativitas seseorang karena tindakan ini tidak membutuhkan tenaga dan tidak harus berfikir keras. Oleh karena itu, tindakan plagiarisme secara perlahan harus di cegah. Dengan memanfaatkan pencocokan string pada dokumen, dapat di kembangkan untuk merancang aplikasi pendeteksi plagiarisme. Metode pencocokan *string* bermacam-macam antara lain *Needlemen Wusnch Algorithm*, *Smith Watermen Algorithm*, *Dynamic Programming Algorithm*, *Brute Force Algorithm*, *Fuzzy String Algorithm* dan *Rabin-Karp Algorithm*. Algoritma ini mempunyai kelebihan dan kekurangannya masing-masing dalam proses *string matching pattern*. Teks pembandingan *string matching pattern* di bagi menjadi dua, yaitu *single string matching pattern* dan *multiple string matching pattern*. Untuk membuat sistem pendeteksi plagiarisme yang baik, memerlukan algoritma yang baik untuk jenis *multiple string matching pattern*.

Algoritma yang cocok untuk permasalahan *multiple string matching pattern* adalah algoritma *Rabin-Karp*.

Algoritma *Rabin-Karp* adalah suatu algoritma pencarian *string* yang ditemukan oleh Michael Rabin dan Richard Karp. Algoritma ini menggunakan *Hashing* untuk menemukan sebuah *substring* dalam sebuah teks. *Hashing* adalah metode menggunakan fungsi *hash* untuk mengubah suatu jenis data menjadi beberapa bilangan bulat sederhana. Algoritma *Rabin-Karp* tidak bertujuan menemukan *string* yang cocok dengan string masukan, melainkan menemukan pola (*pattern*) yang sekiranya cocok dengan *string* masukan [2]. *Rabin-Karp* menghasilkan efisiensi waktu yang baik untuk mendeteksi *string* lebih dari satu pola (*multiple string matching pattern*). Hal ini algoritma *Rabin-Karp* dapat dimanfaatkan untuk melakukan pedeteksian terhadap tindak plagiarisme pada abstraksi skripsi.

Pada penelitian ini akan menggunakan algoritma *Rabin-Karp*. Dalam proses *Rabin-Karp* ada tahap *Preprocessing*. Pada tahap *preprocessing* itu sendiri dokumen uji akan melalui beberapa tahap yaitu *case folding*, *tokenizing*, *filtering* dan proses *Stemming*. Dalam proses *Stemming* menggunakan algoritma Nazief-Adriani. Penggunaan algoritma Nazief-Adriani pada proses *Stemming* akan meningkatkan tingkat akurasi untuk dokumen berbahasa Indonesia di banding algoritma *Stemming* lainnya. *Stemming* adalah salah satu cara yang digunakan untuk meningkatkan permforma *information retrieval* dengan cara mentrasformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya yang biasa disebut dengan *root word* [3].

Pada penelitian sebelumnya yang dilakukan Anton Yudhana, dkk pada tahun 2017 yang berjudul *Implementation of Pattern Matching Algorithm for Portable Document Format* yang membahas tentang bagaimana cara kerja Algoritma *Rabin-Karp* untuk mendeteksi plagiarisme pada suatu dokumen, dalam penelitiannya tersebut menyatakan bahwa algoritma ini cocok untuk mengecek plagiarisme dari suatu dokumen melalui proses *K-Gram* dan *Hashing parsing* untuk menemukan kecocokan kata yang sama atau frase dalam dokumen. Penulis mengambil Algoritma *Rabin-Karp* untuk proses perhitungannya [4].

Berdasarkan latar belakang yang telah di uraikan di atas, maka penulis mencoba untuk melakukan penelitian dengan judul “**Implementasi Algoritma Rabin-Karp Dan Stemming Nazief Adriani Untuk Pendeteksi Plagiarisme Abstraksi Skripsi**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas dapat diuraikan, maka dapat dirumuskan rumusan masalahnya, yaitu: Bagaimana mengimplementasikan algoritma *Rabin-Karp* dan *Stemming Nazief-Adriani* agar dapat mendeteksi plagiarisme yang terdapat pada abstraksi skripsi yang di bandingkan dengan database?

1.3 Batasan Masalah

Dalam pembuatn skripsi ini, agar penelitian dapat terfokus untuk mengatasi permasalahan diatas, akan diberi beberapa batasan masalah, yaitu:

1. Aplikasi ini hanya mendeteksi kemiripan teks yang berbentuk abstraksi skripsi.
2. Aplikasi ini hanya bisa memproses file berupa inputan teks yang di ambil dari repository Universitas Amikom Yogyakarta.
3. Data yang digunakan adalah skripsi mahasiswa Universitas Amikom Yogyakarta.
4. Aplikasi ini hanya menguji data berupa teks, tidak menguji data berupa gambar maupun suara.
5. Aplikasi ini tidak memperhatikan kesalahan ejaan atau penulisan pada dokumen, serta tidak memperhatikan sinonim atau persamaan kata.
6. Data yang di uji menggunakan bahasa Indonesia.

1.4 Maksud dan Tujuan Penelitian

1.4.1 Maksud

Maksud dari penelitian ini adalah mengimplementasi algoritma *Rabin-Karp* dan *Stemming* Nazief-Adriani untuk mendeteksi plagiarisme.

1.4.2 Tujuan

Sedangkan tujuan yang akan di capai dalam penelitian ini adalah untuk mendeteksi kemiripan teks skripsi yang di bandingkan dengan data abstraksi skripsi yang ada di database.

1.5 Manfaat Penelitian

Penelitian ini di harapkan dapat mendeteksi kemiripan teks abstraksi skripsi mahasiswa untuk mengantisipasi plagiarisme.

1.6 Metode Penelitian

Pada penelitian ini, penulis menggunakan beberapa metode penelitian. Adapun metode-metode penelitian yang digunakan adalah sebagai berikut:

1.6.1 Studi Literatur

Menghimpun dan mempelajari teroi-teori mengenai plagiarisme, algoritma *Stemming* Nazief-Adriani, algoritma *Rabin-Karp*. Sumber bacaan berupa *e-book*, jurnal-jurnal, artikel yang di peroleh dari internet guna menunjang penelitian.

1.6.2 Metode Analisis

Data maupun informasi yang didapatkan pada proses pengumpulan data selanjutnya dipelajari dan melakukan analisa tentang algoritma *Rabin-Karp* dan *Stemming* Nazief-Adriani untuk mengembangkan model yang dapat untuk mendeteksi plagiarisme. Agar model yang di usulkan diketahui maka diperlukan implementasi dari model ke bentuk program komputer. Metode analisis yang akan digunakan adalah analisis fungsional dan non fungsional. Selanjutnya hasil dari analisis fungsional dimodelkan dalam bentuk pemodelan *Object Oriented Analysis*(OOA) untuk mengidentifikasi objek-objek yang mendefinisikan tingkah laku sistem. OOA adalah metode analisis yang memeriksa *requirement* (syarat) yang harus dipenuhi dari sebuah sistem dari

sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan.

1.6.3 Metode Perancangan

Perancangan sistem dengan menerapkan algoritma *Stemming* Nazief-Adriani sebagai algoritma *Stemmer*, mengubah kebentuk kata dasar, serta algoritma *Rabin-Karp* sebagai algoritma menyocokkan kata kemudian menghitung persentase kemiripan dokumen. Sistem yang akan dibangun menggunakan metode *Object Oriented Design*(OOD). OOD merupakan teknik perancangan perangkat lunak yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Tools yang dapat digunakan untuk merancang perangkat lunak dengan teknik OOD adalah *Unified Modeling Language* (UML).

1.6.4 Metode Pengembangan

Sistem akan dibangun menggunakan model pengembangan perangkat lunak *Object Oriented Analysis & Design* (OOAD) Life Cycle. OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek. Tahapan dalam OOAD Lyfe Cycle adalah *Project planning*, Analisis Sistem, membuat UML (*Use Case diagram*, *Sequence diagram*, *Activity diagram*), memecah masalah, mendefinisikan kebutuhan, desain awal, implementasi, *deliverable* dan *maintenance*.

1.6.5 Metode Testing

Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan kesalahan. Pengujian perangkat lunak disebut baik apabila pengujian perangkat lunak tersebut dapat menemukan sebuah kesalahan yang tidak terungkap. Proses pengujian menentukan kelayakan dari sistem yang

dibangun. Tujuan utama dari pengujian adalah untuk menemukan kesalahan dan fungsi dari perangkat lunak yang tidak sesuai dengan tujuan pengembangan yang secara sistematis. Pada penelitian ini, metode pengujian yang digunakan adalah *White Box* dan *Black Box*. Proses pengujian akan dilakukan terhadap semua kebutuhan fungsional yang telah dirancang pada tahap perancangan sistem, guna menemukan kesalahan yang mungkin terjadi pada program yang dibangun, agar program tersebut sesuai dengan kebutuhan fungsionalnya.

1.7 Sistematika Penulisan

Penulisan skripsi ini disusun secara sistematis kedalam beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisikan mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, maksud dan tujuan dari pembahasan, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini membahas teori-teori pendukung yang berkaitan dengan skripsi yang akan dijadikan dasar penelitian ini. Teori yang akan diangkat yaitu mengenai deteksi plagiarisme dengan algoritma *Rabin-Karp* dan *Stemming Nazief-Adriani*.

BAB III ANALISIS DAN PERANCANGAN

Bab ini menjabarkan tentang tujuan dari perancangan sistem, analisa kebutuhan sistem, perancangan perangkat lunak dan juga tahapan dalam mengimplementasikan metode.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini akan membahas mengenai implementasi sistem pendeteksi tingkat plagiarisme abstraksi skripsi menggunakan algoritma *Rabin-Karp* dan *Stemming* Nazief-Adriani serta di sajikan evaluasi dan kesimpulan dari pengujian yang telah dilakukan terhadap sistem.

BAB V PENUTUP

Bab ini berisi kesimpulan dari skripsi yang dibuat dan menjelaskan saran-saran penulis kepada pembaca agar penerapan algoritma *Rabin-Karp* dan *Stemming* Nazief-Adriani dan diharapkan dapat bermanfaat dalam pengembangan lebih lanjut.

DAFTAR PUSTAKA

Berisi sumber bacaan yang penulis gunakan sebagai bahan penelitian.

