

ABSTRAK

IMPLEMENTASI SISTEM DETEKSI PLAGIARISME DOKUMEN MENGGUNAKAN COSINE SIMILARITY BERBASIS GUI DENGAN PYTHON DAN TKINTER

Nama : Kristika Meza Werda

NPM : 2155201147

Pembimbing : Khairullah, S.T., M.Kom

Penelitian ini bertujuan merancang dan mengimplementasikan sistem deteksi plagiarisme dokumen berbasis desktop yang dapat dioperasikan secara offline menggunakan algoritma Cosine Similarity dengan pembobotan TF-IDF (Term Frequency-Inverse Document Frequency), dibangun menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI) berbasis Tkinter. Sistem mengimplementasikan proses pra-pemrosesan teks yang mencakup *case folding*, *cleaning*, *tokenization*, dan *stopword removal* menggunakan ekspresi reguler Python, dilanjutkan dengan pembobotan TF-IDF untuk merepresentasikan dokumen sebagai vektor numerik, kemudian menghitung kemiripan antar dokumen menggunakan formula Cosine Similarity dengan rentang skor 0 hingga 1. Hasil skor dikategorikan ke dalam empat label yaitu Tidak Plagiat, Plagiarisme Rendah, Plagiarisme Sedang, dan Plagiarisme Tinggi. Aplikasi dibangun dengan struktur navigasi tiga tab meliputi tab Deteksi Plagiarisme, tab Pengujian SUS, dan tab Pengujian Black Box, serta dilengkapi fitur ekspor laporan ke format Microsoft Word dan Excel. Pengujian fungsionalitas menggunakan Black Box Testing terhadap 20 skenario uji menghasilkan tingkat keberhasilan sebesar 95% (19 dari 20 kasus uji PASS). Satu kasus uji yang dinyatakan FAIL disebabkan oleh kekeliruan pada rancangan data uji, bukan pada kesalahan logika program. Pengujian usability menggunakan System Usability Scale (SUS) terhadap 10 responden menghasilkan rata-rata skor sebesar 80,50 yang termasuk dalam kategori "Good", dengan 70% responden memberikan penilaian pada kategori Good hingga Excellent. Hasil penelitian menunjukkan bahwa sistem berhasil membedakan dokumen identik (skor 1,0000), dokumen hasil parafrase sebagian (skor 0,6954), dan dokumen tidak berkaitan (skor 0,0000) secara akurat, serta dapat dioperasikan sepenuhnya secara offline tanpa bergantung pada pustaka pembelajaran mesin eksternal.

Kata Kunci: Deteksi Plagiarisme, Cosine Similarity, TF-IDF, GUI, Python, Tkinter, Black Box Testing, System Usability Scale

ABSTRACT

IMPLEMENTATION OF A DOCUMENT PLAGIARISM DETECTION SYSTEM USING COSINE SIMILARITY WITH A GUI-BASED APPROACH IN PYTHON AND TKINTER

Name : Kristika Meza Werda

NPM : 2155201147

Advisor : Khairullah, S.T., M.Kom

This study aims to design and implement a desktop-based document plagiarism detection system that can be operated offline using the Cosine Similarity algorithm with TF-IDF (Term Frequency-Inverse Document Frequency) weighting, built using the Python programming language with a Tkinter-based Graphical User Interface (GUI). The system implements a text preprocessing pipeline comprising *case folding*, *cleaning*, *tokenization*, and *stopword removal* using Python regular expressions, followed by TF-IDF weighting to represent documents as numerical vectors, and then calculates document similarity using the Cosine Similarity formula with a score range of 0 to 1. The resulting scores are categorized into four labels: Not Plagiarism, Low Plagiarism, Moderate Plagiarism, and High Plagiarism. The application is built with a three-tab navigation structure covering the Plagiarism Detection tab, the SUS Testing tab, and the Black Box Testing tab, and is equipped with a report export feature to Microsoft Word and Excel formats. Functional testing using Black Box Testing on 20 test scenarios yielded a success rate of 95% (19 out of 20 test cases PASS). The one FAIL test case was caused by an error in the test data design rather than a program logic error. Usability testing using the System Usability Scale (SUS) involving 10 respondents produced an average score of 80.50, classified as "Good", with 70% of respondents rating the system in the Good to Excellent category. The results demonstrate that the system accurately distinguishes between identical documents (score 1.0000), partially paraphrased documents (score 0.6954), and unrelated documents (score 0.0000), and can be operated fully offline without relying on external machine learning libraries.

Keywords: Plagiarism Detection, Cosine Similarity, TF-IDF, GUI, Python, Tkinter, Black Box Testing, System Usability Scale

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia akademik berdiri di atas pondasi kejujuran intelektual. Setiap karya tulis ilmiah yang lahir dari proses berpikir serius adalah cerminan dari kerja keras penulisnya, dan karena itulah keaslian sebuah tulisan menjadi sesuatu yang tidak bisa dinegosiasikan. Sayangnya, realita yang terjadi di lapangan tidak selalu sejalan dengan harapan tersebut. Plagiarisme masih menjadi persoalan yang terus membayangi institusi pendidikan, fenomena ini tidak hanya mempengaruhi kualitas pendidikan, tetapi juga mengancam reputasi akademik dan kredibilitas lulusan dari institusi pendidikan tinggi. Lebih jauh lagi, pada tingkat individu, mahasiswa yang melakukan plagiarisme berisiko merusak reputasi akademiknya, kehilangan kepercayaan dari pembimbing, komite penguji, dan rekan sejawat, yang pada akhirnya dapat menghambat peluang kolaborasi penelitian di masa depan (Amalina and Ardiansyah, 2025).

Masalah utama yang ingin diangkat dalam penelitian ini yaitu mengenai pemeriksaan kemiripan dokumen khususnya karya ilmiah mahasiswa seperti skripsi, laporan, dan tugas akhir masih sangat bergantung pada alat komersial seperti Turnitin yang membutuhkan koneksi internet dan berbayar. Kondisi ini menjadi hambatan tersendiri bagi institusi pendidikan, terutama yang berada di daerah dengan akses internet yang tidak stabil, atau bagi mahasiswa yang tidak memiliki

kemampuan finansial untuk mengakses alat tersebut. Kebutuhan akan sebuah sistem deteksi plagiarisme yang bisa dijalankan secara lokal (offline), mudah digunakan, dan tidak berbayar menjadi sangat relevan untuk diteliti dan dikembangkan.

Sejumlah penelitian terdahulu telah mencoba menjawab persoalan ini dari berbagai sudut pandang. (Fadhullah, Fauziah and Winarsih, 2022) membangun aplikasi deteksi dini plagiarisme pada penelitian ilmiah menggunakan algoritma Cosine Similarity berbasis web, dengan tujuan memudahkan mahasiswa dalam menentukan judul tugas akhir agar terhindar dari potensi kesamaan judul yang sudah ada sebelumnya. Selanjutnya, (Lumbansiantar, Dwiasnati and Fatonah, 2023) dalam penelitiannya menunjukkan bahwa metode Cosine Similarity mampu mendeteksi plagiarisme pada dokumen jurnal dengan cara menghitung persentase kemiripan antara dokumen repositori dengan dokumen uji. Lebih lanjut, (Darmanto, Indah Pradasari and Wahyudi, 2024) mengembangkan sistem deteksi plagiarisme tugas akhir mahasiswa berbasis Natural Language Processing menggunakan kombinasi algoritma Jaro-Winkler dan TF-IDF, dengan data pengujian yang diambil dari 20 dokumen tugas akhir mahasiswa Jurusan Teknik Elektro dan Teknik Informatika.

Meskipun penelitian-penelitian tersebut sudah memberikan kontribusi yang berarti, terdapat beberapa kesenjangan yang masih belum terjawab. Pertama, hampir seluruh sistem yang dikembangkan berbasis web, sehingga tetap memerlukan koneksi internet dan infrastruktur server untuk bisa berjalan. Kedua, belum banyak penelitian yang secara khusus membangun sistem deteksi

plagiarisme berbasis antarmuka grafis (Graphical User Interface/GUI) yang berdiri sendiri sebagai aplikasi desktop dan dapat dioperasikan sepenuhnya secara offline. (Arkan *et al.*, 2026) memang mengembangkan aplikasi deteksi plagiarisme dengan TF-IDF dan Cosine Similarity yang dirancang beroperasi tanpa koneksi internet, namun implementasinya masih mengacu pada pendekatan yang belum memanfaatkan antarmuka grafis berbasis Python Tkinter secara penuh. Celah inilah yang menjadi titik berangkat penelitian ini.

Berdasarkan latar belakang dan kesenjangan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi plagiarisme dokumen yang memanfaatkan metode Cosine Similarity dengan pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*), yang dikemas dalam bentuk aplikasi desktop berbasis GUI menggunakan bahasa pemrograman Python dan library Tkinter sebagai komponen antarmukanya. Sistem ini dirancang untuk mampu membandingkan dua atau lebih dokumen teks, menghitung nilai kemiripan antar dokumen, dan menyajikan hasilnya dalam tampilan yang ramah pengguna tanpa membutuhkan koneksi internet sama sekali.

Metode yang diusulkan dalam penelitian ini mengacu pada proses yang terstruktur. Pertama, dokumen yang akan dianalisis akan melalui tahap *preprocessing* yang meliputi *case folding*, *tokenization*, *stopword removal*, dan *stemming* untuk membersihkan teks dari elemen yang tidak relevan. Selanjutnya, setiap term dalam dokumen akan dihitung bobotnya menggunakan TF-IDF untuk merepresentasikan kepentingan kata dalam konteks keseluruhan dokumen. Cosine Similarity kemudian digunakan untuk mengukur kemiripan antar dokumen

berdasarkan representasi vektor yang telah terbentuk, dengan membandingkan sudut antara dua vektor dalam ruang multidimensi. Hasil perhitungan ini kemudian ditampilkan secara visual melalui antarmuka Tkinter yang telah dirancang agar intuitif dan mudah dioperasikan (Asmaul Husnah Nasrullah, 2024).

Adapun manfaat yang diharapkan dari penelitian ini mencakup dua dimensi. Secara praktis, sistem ini diharapkan dapat menjadi alat bantu yang nyata bagi mahasiswa, dosen, dan tenaga kependidikan dalam memeriksa tingkat kemiripan dokumen karya ilmiah sebelum diserahkan atau dipublikasikan, sehingga budaya akademik yang jujur dan bertanggung jawab dapat lebih mudah diterapkan. Sinergi antara pemerintah, lembaga pendidikan, dan penelitian diperlukan untuk memperkuat sistem deteksi dan sosialisasi mengenai pentingnya integritas akademik — dan penelitian ini merupakan salah satu bentuk kontribusi nyata dari ranah teknologi informasi terhadap upaya tersebut. Secara teoretis, penelitian ini diharapkan dapat memperkaya khazanah keilmuan di bidang *Natural Language Processing* dan pengembangan perangkat lunak, khususnya dalam konteks pengembangan sistem deteksi kemiripan teks berbasis algoritma yang dapat diimplementasikan secara lokal dan mandiri.

1.2 Pertanyaan Penelitian

1. Bagaimana merancang dan mengimplementasikan proses pra-pemrosesan teks (*case folding*, *tokenization*, *stopword removal*, dan *stemming* menggunakan Sastrawi) untuk mempersiapkan dokumen masukan sebelum proses perhitungan kemiripan?

2. Bagaimana mengukur tingkat kemiripan antar dokumen menggunakan algoritma Cosine Similarity dan menentukan kategori tingkat plagiarisme berdasarkan nilai skor yang dihasilkan?
3. Bagaimana merancang antarmuka GUI berbasis Python Tkinter yang intuitif dan mudah digunakan agar pengguna dapat melakukan deteksi plagiarisme tanpa memerlukan koneksi internet?

1.3 Tujuan Penelitian

1. Mengimplementasikan proses pra-pemrosesan teks yang mencakup *case folding*, *tokenization*, *stopword removal*, dan *stemming* menggunakan library Sastrawi untuk mempersiapkan dokumen masukan sebelum proses perhitungan kemiripan dilakukan
2. Mengimplementasikan algoritma Cosine Similarity untuk mengukur tingkat kemiripan antar dokumen dan mengklasifikasikannya ke dalam kategori tingkat plagiarisme berdasarkan nilai skor yang dihasilkan.
3. Merancang dan membangun antarmuka grafis (*Graphical User Interface*) berbasis Python Tkinter yang intuitif dan mudah dioperasikan tanpa memerlukan koneksi internet.

1.4 Kerangka Kerja Penelitian (*Research Framework*)

Framework penelitian merupakan kerangka kerja sistematis yang menggambarkan seluruh tahapan pelaksanaan penelitian secara terstruktur dan terukur dari awal hingga akhir. Pada penelitian ini, framework disusun dengan mengintegrasikan dua lapisan pendekatan yang saling melengkapi, yaitu model

pengembangan sistem *Waterfall* sebagai kerangka makro yang mengatur urutan fase pembangunan aplikasi, dan alur kerja algoritma deteksi plagiarisme sebagai inti teknis yang menjelaskan proses komputasional di dalam sistem.

Tabel 1. Framework Penelitian

No	Fase Waterfall	Tahap Penelitian	Kegiatan Utama	Output yang Dihasilkan
1	Requirements Analysis	Studi Literatur & Analisis Kebutuhan	Mengkaji teori plagiarisme, Cosine Similarity, TF-IDF, NLP, GUI Tkinter, dan penelitian terdahulu; mengidentifikasi kebutuhan fungsional dan non-fungsional sistem	Dokumen kebutuhan sistem (Software Requirements Specification) dan landasan teori penelitian
2	Requirements Analysis	Identifikasi Masalah	Menganalisis permasalahan ketergantungan pada alat deteksi plagiarisme berbasis web yang membutuhkan internet dan biaya berlangganan	Rumusan masalah teknis yang menjadi dasar perancangan sistem
3	System Design	Perancangan Sistem	Merancang Diagram Konteks, Data Flow Diagram (DFD) Level 0, dan Flowchart alur kerja algoritma deteksi plagiarisme	Artefak desain sistem: Diagram Konteks, DFD, dan Flowchart algoritma
4	System Design	Perancangan Antarmuka	Merancang wireframe antarmuka GUI Tkinter untuk tiga tab utama (Deteksi Plagiarisme, Pengujian SUS, Pengujian Black Box)	Wireframe rancangan antarmuka sistem (hitam putih)
5	Implementation	Pra-pemrosesan Teks	Mengimplementasikan proses case folding, cleaning, tokenization, dan stopwords removal menggunakan ekspresi reguler Python	Modul pra-pemrosesan teks yang menghasilkan kumpulan token kata bersih
6	Implementation	Pembobotan TF-IDF	Mengimplementasikan metode Term Frequency-Inverse Document Frequency secara native menggunakan pustaka matematika bawaan Python	Representasi vektor TF-IDF untuk setiap dokumen yang dibandingkan

No	Fase Waterfall	Tahap Penelitian	Kegiatan Utama	Output yang Dihasilkan
7	Implementation	Implementasi Cosine Similarity	Mengimplementasikan formula Cosine Similarity untuk menghitung skor kemiripan antar dua dokumen (rentang 0–1)	Nilai skor dan persentase kemiripan antar pasangan dokumen
8	Implementation	Pengembangan GUI Tkinter	Membangun antarmuka pengguna grafis menggunakan pustaka Tkinter dengan struktur tiga tab navigasi (Notebook)	Aplikasi desktop berbasis GUI yang dapat dioperasikan secara offline
9	Implementation	Modul Ekspor Laporan	Mengimplementasikan fitur ekspor hasil deteksi, data SUS, dan hasil Black Box ke format Microsoft Word dan Excel	Fitur ekspor laporan (.docx dan .xlsx) yang berfungsi pada ketiga tab
10	Testing	Pengujian Black Box	Menjalankan 20 skenario pengujian fungsional otomatis terhadap fungsi inti sistem (preprocessing, TF-IDF, Cosine Similarity, kategorisasi)	Laporan hasil pengujian Black Box yang menunjukkan status PASS/FAIL tiap skenario
11	Testing	Pengujian System Usability Scale	Mengumpulkan jawaban kuesioner SUS dari sejumlah responden setelah mencoba sistem secara langsung	Skor SUS per responden dan rekapitulasi skor keseluruhan beserta interpretasinya
12	Maintenance	Analisis Hasil & Perbaikan	Menganalisis hasil pengujian Black Box dan SUS, memperbaiki bug serta menyempurnakan antarmuka berdasarkan temuan	Sistem yang telah disempurnakan beserta catatan hasil pengujian
13	Maintenance	Penyusunan Laporan	Mendokumentasikan seluruh proses penelitian secara sistematis mulai dari pendahuluan hingga kesimpulan dan saran	Laporan skripsi lengkap (Bab I sampai dengan Bab V)

Framework penelitian pada Tabel 1 disusun secara sistematis dengan mengintegrasikan model Waterfall sebagai kerangka pengembangan sistem dengan kegiatan teknis nyata yang telah dilaksanakan pada setiap tahapnya. Penelitian

dimulai dengan studi literatur dan identifikasi masalah pada fase Requirements Analysis, dilanjutkan dengan perancangan seluruh artefak desain sistem pada fase System Design mencakup Diagram Konteks, DFD, Flowchart algoritma, dan wireframe antarmuka GUI.

Fase Implementation merupakan fase terpanjang dan terkompleks, mencakup pengkodean modul pra-pemrosesan teks, pembobotan TF-IDF, perhitungan Cosine Similarity, pembangunan antarmuka GUI Tkinter yang terintegrasi dalam tiga tab utama, serta implementasi fitur ekspor laporan ke format Word dan Excel. Fase Testing kemudian memverifikasi bahwa seluruh fitur berfungsi sesuai kebutuhan melalui Black Box Testing, sekaligus mengukur tingkat kegunaan sistem dari sisi pengguna melalui pengujian System Usability Scale (SUS).

Terakhir, fase Maintenance melakukan penyempurnaan sistem berdasarkan temuan-temuan pada fase pengujian serta menyusun seluruh dokumentasi penelitian dalam bentuk laporan skripsi yang lengkap dan sistematis. Dengan framework yang terstruktur ini, penelitian diharapkan mampu menghasilkan aplikasi desktop deteksi plagiarisme yang akurat, fungsional, mudah digunakan, dan dapat dioperasikan sepenuhnya secara offline.