

IMPLEMENTASI DETEKSI PLAGIARISME CODE DENGAN METODE WINNOWER, DAN JACCARD SIMILARITY UNTUK PEMERIKSAAN TUGAS LABORATORIUM INFORMATIKA

Muhammad Dasril Asdar*¹, Lukman², Muhyiddin AM Hayat³

^{1,2,3}Informatika, Universitas Muhamadiyah Makassar, Makassar, 90221, Indonesia

e-mail koresponden:

105841100321@student.unismuh.ac.id¹, lukman@unismuh.ac.id², muhyiddin@student.unismuh.ac.id³.

Received: Februari 01,2026; Accepted: Maret 01, 2026; Published: Maret 31, 2026

Abstrak

Integritas akademik dalam tugas pemrograman sering kali terganggu oleh praktik plagiarisme yang sulit dideteksi secara manual, terutama ketika terjadi modifikasi pada nama variabel atau struktur kode. Penelitian ini bertujuan untuk mengimplementasikan sistem deteksi plagiarisme kode sumber berbasis web yang mengintegrasikan algoritma Winnower untuk pembentukan fingerprint dokumen dan Jaccard Similarity untuk pengukuran tingkat kemiripan. Sistem dirancang dengan tahapan pre-processing khusus yang meliputi tokenisasi dan normalisasi identifier untuk memastikan deteksi berfokus pada logika program, bukan sekadar kesamaan sintaksis. Pengujian dilakukan menggunakan corpus tugas laboratorium informatika dengan tiga skenario utama: modifikasi minor, perbedaan bahasa pemrograman, dan perubahan nama variabel (identifier). Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi kemiripan sebesar 68% pada modifikasi minor dan mempertahankan tingkat deteksi sebesar 61,54% pada kode yang mengalami perubahan nama variabel secara menyeluruh. Selain itu, sistem menunjukkan akurasi tinggi dengan hasil 0% (tanpa false positive) pada pengujian lintas bahasa dan memiliki tingkat konsistensi algoritma 100%. Hasil ini membuktikan bahwa integrasi Winnower dengan normalisasi identifier efektif sebagai solusi verifikasi keaslian tugas pemrograman yang andal dan objektif.

Kata kunci: Deteksi Plagiarisme, Winnower, Jaccard Similarity, Normalisasi Kode, Source Code

Abstract

Academic integrity in programming assignments is often compromised by plagiarism practices that are difficult to detect manually, especially when variable names or code structures are modified. This study aims to implement a web-based source code plagiarism detection system by integrating the Winnower algorithm for document fingerprinting and Jaccard Similarity for measuring similarity levels. The system is designed with specific pre-processing stages, including tokenization and identifier normalization, to ensure detection focuses on program logic rather than mere syntactic similarity. Testing was conducted using an informatics laboratory assignment corpus with three main scenarios: minor modifications, cross-language comparison, and variable name (identifier) changes. The results show that the system successfully detected 68% similarity in minor modifications and maintained a 61.54% detection rate on code that underwent extensive variable renaming. Furthermore, the system demonstrated high accuracy with 0% (no false positives) in cross-language testing and achieved 100% algorithm consistency. These results prove that integrating Winnower with identifier normalization is effective as a reliable and objective solution for verifying programming assignment authenticity.

Keyword: Plagiarism Detection, Winnower, Jaccard Similarity, Code Normalization, Source Code

1. Pendahuluan

Integritas akademik merupakan pilar fundamental dalam pendidikan tinggi, namun perkembangan teknologi informasi telah membawa tantangan baru berupa kemudahan akses terhadap sumber daya digital. Praktik plagiarisme kini semakin kompleks seiring dengan menjamurnya repositori kode daring seperti GitHub, yang sering digunakan sebagai media kolaborasi pembelajaran [1]. Halim dan Lasut mencatat bahwa plagiarisme di era digital tidak lagi terbatas pada penyalinan teks secara langsung, tetapi juga mencakup penggunaan ulang kode sumber tanpa atribusi yang memadai, yang mencederai keadilan evaluasi akademik [2].

Berbeda dengan karya tulis ilmiah berbentuk narasi, kode sumber (source code) memiliki karakteristik logis-struktural yang unik. Soldati et al. menekankan bahwa dua program dapat memiliki makna fungsional yang identik meskipun ditulis dengan gaya penulisan atau struktur sintaksis yang berbeda [3]. Hal ini dipertegas oleh studi Pradeep et al., yang menyatakan bahwa pengembangan sistem deteksi plagiarisme kode perlu mempertimbangkan pendekatan kombinasi struktur dan sintaksis untuk mengenali kemiripan tingkat lanjut yang mungkin terlewatkan oleh metode deteksi teks biasa [4].

Berbagai metode telah dikembangkan untuk mengatasi permasalahan ini. Eka Putra dalam penelitiannya membangun sistem deteksi menggunakan Cosine Similarity dan pembobotan TF-IDF, yang terbukti efektif untuk analisis berbasis frekuensi token [5]. Namun, pendekatan statistik murni sering kali kurang sensitif terhadap urutan logika program. Sebagai alternatif, algoritma document fingerprinting seperti Winnowing menjadi pilihan populer. Ramli et al. menunjukkan bahwa Winnowing mampu mendeteksi plagiarisme pada tugas pemrograman mahasiswa dengan rata-rata tingkat kemiripan mencapai 75,12% [6]. Efektivitas algoritma ini juga didukung oleh penelitian Rijoly et al., yang menemukannya optimal dalam mendeteksi kemiripan teks akademik dengan konfigurasi parameter k-gram tertentu [7].

Meskipun efektif, algoritma Winnowing memiliki tantangan tersendiri. Arnawa menyoroti pentingnya penggunaan teknik seperti rolling hash untuk meningkatkan presisi deteksi [8]. Tantangan terbesar tetap terletak pada teknik manipulasi permukaan (surface-level modification). Saputra et al. membuktikan bahwa modifikasi seperti penggantian nama variabel (identifikasi renaming), penyisipan komentar, atau perubahan format penulisan sering kali dilakukan untuk mengelabui sistem deteksi [9]. Tanpa penanganan khusus, sistem berisiko menghasilkan false positive, di mana kode dianggap berbeda padahal logikanya sama, atau sebaliknya [10].

Kelemahan utama dari banyak pendekatan sebelumnya adalah kurangnya tahapan pre-processing yang mendalam untuk menormalisasi variasi kode tersebut. Bramantya et al. menegaskan bahwa tahapan pre-processing seperti normalisasi identifier dan pembentukan k-gram sangat krusial untuk memastikan bahwa kemiripan logika program tetap terdeteksi meskipun terjadi penyamaran sintaksis [11]. Selain akurasi algoritma, aspek efisiensi sistem juga penting. Gea dan Susetyo menyatakan bahwa implementasi sistem berbasis web menggunakan framework ringan seperti Flask dapat mempercepat waktu pembangunan dan respons sistem, yang sangat dibutuhkan dalam lingkungan laboratorium dengan volume tugas yang besar [12].

Berdasarkan paparan tersebut, penelitian ini bertujuan untuk mengimplementasikan sistem deteksi plagiarisme kode sumber berbasis web yang mengintegrasikan algoritma Winnowing dengan metode Jaccard Similarity [13]. Kebaruan penelitian ini terletak pada penerapan teknik pre-processing komprehensif berupa normalisasi identifier untuk menyamakan token kode ke dalam bentuk umum sebelum proses fingerprinting. Pendekatan ini diharapkan mampu menghasilkan verifikasi yang tidak hanya akurat dan objektif, tetapi juga tahan terhadap upaya modifikasi sintaksis minor yang sering dilakukan mahasiswa..

2. Metode Penelitian

2.1. Alat dan Bahan

Alat dan bahan yang digunakan terdiri dari perangkat keras dan perangkat lunak sebagai berikut.

a. Perangkat Keras

- Komputer dengan Prosesor Intel Core i5-10400F
- Kartu Grafis (GPU) AMD Radeon RX580
- Memori (RAM) sebesar 16 GB
- Media penyimpanan SSD 1256 GB

Perangkat keras tersebut digunakan untuk menjalankan proses komputasi algoritma Winnowing, pengelolaan basis data corpus, serta pengujian performa sistem secara lokal.

b. Perangkat Lunak

- Sistem operasi Windows
- Visual Studio Code sebagai code editor
- Bahasa pemrograman Python (Flask) untuk backend
- Library Git dan GitHub untuk manajemen data repositori
- Library Hash dan String untuk pemrosesan teks kode

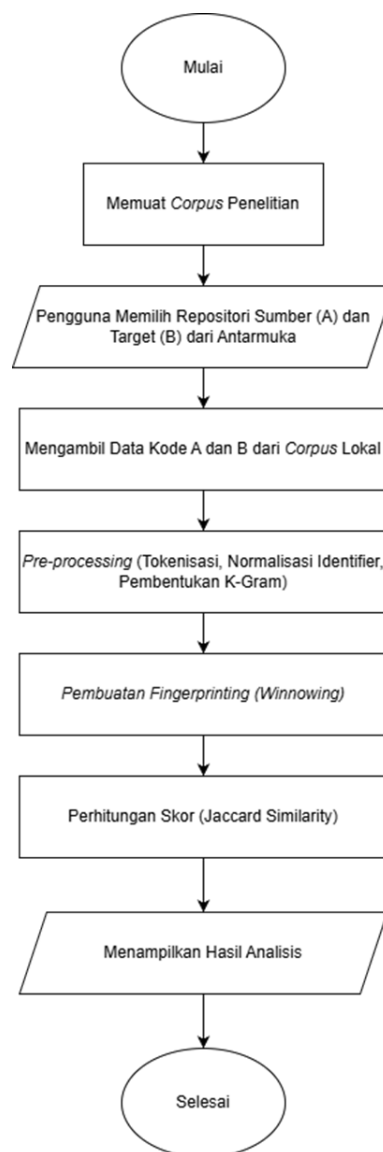
Seluruh perangkat lunak tersebut digunakan untuk mengimplementasikan algoritma Winnowing dan Jaccard Similarity, membangun sistem deteksi plagiarisme berbasis web, serta melakukan pengujian fungsional sistem.

2.2. Perancangan Sistem

Perancangan sistem dilakukan untuk membangun mekanisme deteksi plagiarisme kode yang efisien dalam memeriksa tugas pemrograman mahasiswa. Sistem dirancang untuk memproses repositori kode melalui tahapan pre-processing (tokenisasi dan normalisasi) sebelum dilakukan pencocokan fingerprint.

Alur kerja sistem dimulai dari proses pemuatan corpus penelitian dan pemilihan repositori sumber serta target. Selanjutnya, sistem melakukan proses pembersihan (cleaning) dan normalisasi identifier pada kode program, kemudian memecahnya menjadi token-token k-gram. Nilai hash dari k-gram tersebut digunakan sebagai input dalam proses pembentukan fingerprint dokumen menggunakan algoritma *Winnowing*.

Setelah proses pembentukan fingerprint selesai, sistem menghitung tingkat kemiripan menggunakan metode *Jaccard Similarity*. Alur keseluruhan proses sistem ini digambarkan secara ringkas pada Gambar 1. Flowchart Pelaksanaan Sistem. Gambar tersebut menunjukkan hubungan antar tahapan mulai dari input repositori, proses normalisasi dan fingerprinting, hingga visualisasi hasil kemiripan.



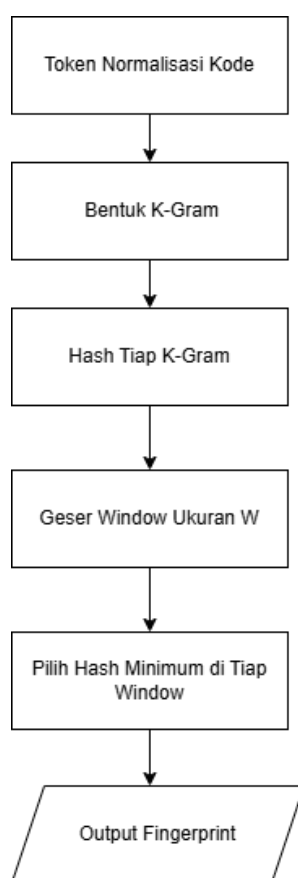
Gambar 1. Flowcart Pelaksanaan Sistem

2.3. Implementasi Algoritma WInnowing

Implementasi algoritma WInnowing pada penelitian ini digunakan sebagai metode document fingerprinting untuk memilih representasi unik dari kode program. Algoritma ini terdiri dari tiga tahap utama dalam konteks sistem ini, yaitu pembentukan k-gram, hashing, dan pemilihan fingerprint melalui windowing.

Pada tahap pembentukan k-gram, sistem memecah urutan token hasil normalisasi menjadi potongan-potongan dengan panjang tetap ($k=5$). Selanjutnya, setiap k-gram dikonversi menjadi nilai hash numerik. Tahap pemilihan fingerprint dilakukan dengan menerapkan jendela geser (sliding window) berukuran $w=10$ pada deretan hash tersebut. Dari setiap jendela, sistem memilih satu nilai hash minimum sebagai fingerprint.

Pendekatan ini bertujuan untuk mengurangi jumlah data yang perlu dibandingkan namun tetap mempertahankan karakteristik unik dokumen, sehingga sistem tetap mampu mendeteksi kemiripan meskipun terjadi perubahan minor. Tahapan algoritma WInnowing yang digunakan dalam penelitian ini divisualisasikan pada Gambar 2. Flowchart Algoritma WInnowing.



Gambar 2. Flowchart Algoritma WInnowing

2.4. Perhitungan Kemiripan (Jaccard Similarity)

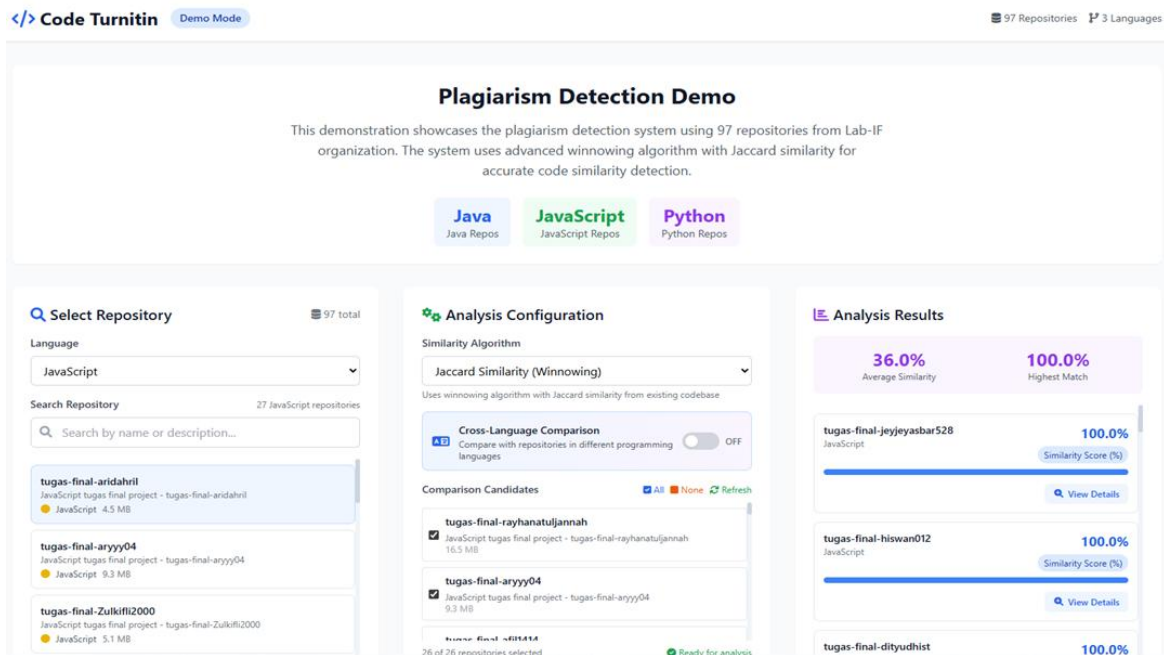
Perhitungan kemiripan diterapkan untuk memberikan nilai kuantitatif terhadap tingkat kesamaan antar dua berkas kode. Metode yang digunakan adalah Jaccard Similarity, yang menghitung rasio antara jumlah fingerprint yang sama (irisan) dengan total fingerprint unik (gabungan) dari kedua dokumen.

Rumus yang diterapkan adalah membagi jumlah irisan himpunan fingerprint A dan B dengan jumlah gabungan himpunan keduanya. Hasil perhitungan ini kemudian dikonversi ke dalam bentuk persentase (0-100%). Dokumen dinyatakan memiliki indikasi plagiarisme tinggi apabila persentase kemiripan mendekati 100%, yang berarti struktur logika dan token penyusunnya sangat identik.

2.5. Implementasi Antarmuka Sistem

Antarmuka sistem dibangun berbasis web untuk memudahkan interaksi pengguna dalam melakukan pemeriksaan tugas. Halaman antarmuka dirancang untuk menampilkan daftar repositori corpus, konfigurasi analisis, serta tabel hasil deteksi kemiripan secara terintegrasi.

Sistem menampilkan hasil analisis yang memuat skor persentase kemiripan dan menyediakan fitur visualisasi side-by-side untuk membandingkan blok kode yang terdeteksi mirip. Implementasi antarmuka sistem ditunjukkan pada Gambar 3. Tampilan Awal Sistem Deteksi Plagiarisme, yang memperlihatkan halaman utama aplikasi.



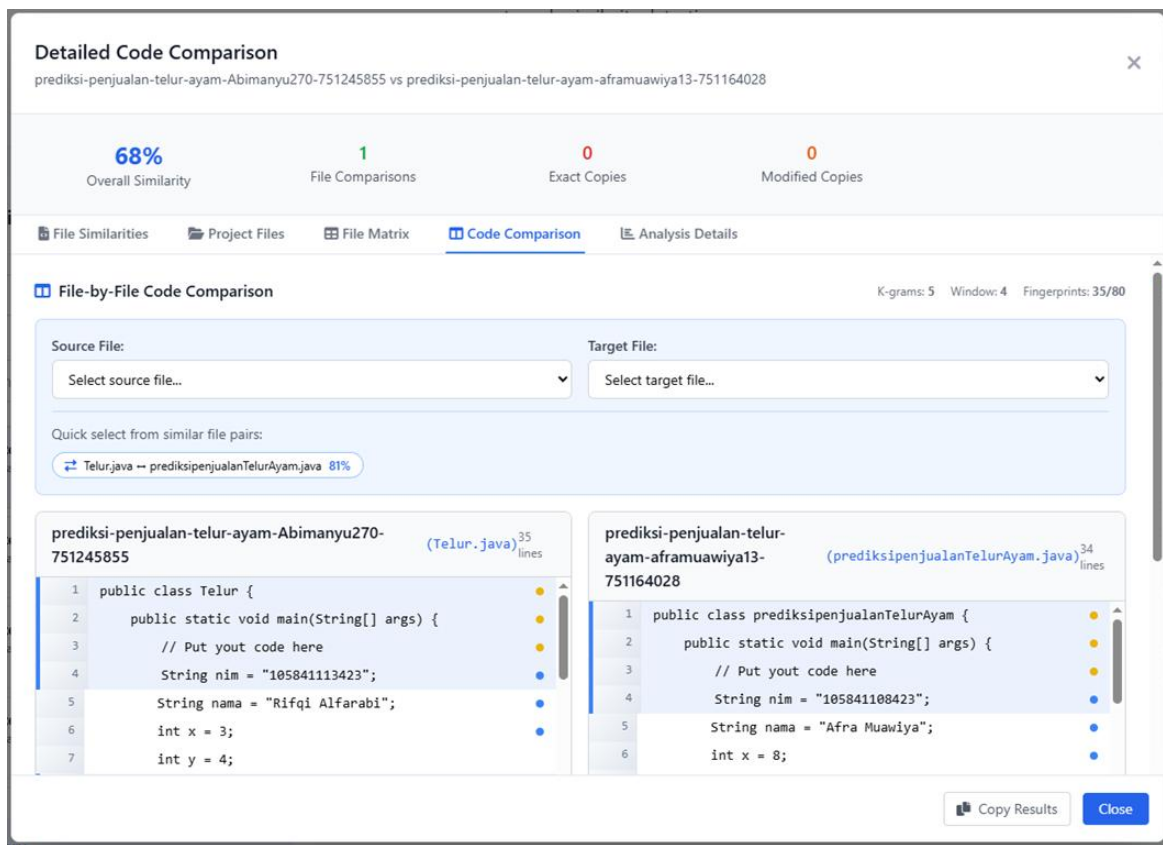
Gambar 3. Tampilan Awal Sistem Deteksi Plagiarisme

3. Hasil dan Pembahasan

3.1. Implementasi Sistem Deteksi Plagiarisme

Hasil implementasi menunjukkan bahwa sistem deteksi plagiarisme kode sumber berhasil dikembangkan sesuai dengan perancangan arsitektur layered yang diusulkan. Sistem dibangun menggunakan framework Flask pada sisi backend dan Tailwind CSS pada sisi frontend, menghasilkan antarmuka berbasis web yang responsif.

Sistem mampu memproses corpus repositori, menjalankan normalisasi kode, dan menampilkan hasil perhitungan kemiripan secara akurat. Antarmuka utama menampilkan daftar repositori dan konfigurasi analisis, sebagaimana ditunjukkan pada Gambar 4. Tampilan Hasil Analisis Kemiripan. Gambar tersebut memperlihatkan hasil deteksi kemiripan antara dua repositori dengan skor persentase yang spesifik.



Gambar 4. Tampilan Hasil Analisis Kemiripan

3.2. Hasil Pengujian Akurasi Deteksi

Pengujian akurasi dilakukan untuk mengukur efektivitas algoritma WInnowing dan teknik normalisasi identifier dalam berbagai kondisi. Pengujian mencakup tiga skenario: modifikasi minor, perbedaan bahasa pemrograman (cross-language), dan perubahan nama variabel (identifier renaming).

Hasil pengujian dirangkum pada Tabel 1. Berdasarkan tabel tersebut, sistem berhasil mendeteksi kemiripan sebesar 68% pada kode dengan modifikasi minor dan mempertahankan deteksi sebesar 61,54% pada kode yang mengalami perubahan nama variabel total. Hal ini membuktikan bahwa tahap pre-processing berhasil mempertahankan pola logika program meskipun sintaksisnya diubah.

Tabel 1. Hasil Pengujian Akurasi Sistem

Skenario Pengujian	Repository Sumber vs Pembanding	Skor Kemiripan	Keterangan Hasil
Modifikasi Minor	Abimanyu270(Java) vs Aframuawiya 13 (Java)	68,0%	Pola logika utama berhasil terdeteksi meskipun ada perubahan kecil pada struktur.
Lintas Bahasa	Abimanyu270(Java) vs Rayhanatul (JS)	0,0%	Sistem akurat, tidak menghasilkan deteksi palsu (<i>false positive</i>) pada bahasa pemrograman berbeda.
Normalisasi Identifier	Abimanyu270(Asli) vs AliRahmat (Modifikasi)	61,54%	Efektivitas normalisasi terbukti; sistem tetap mengenali logika meski nama variabel diubah total.

3.3. Analisis Konsistensi Algoritma

Pengujian konsistensi dilakukan untuk memastikan reliabilitas sistem dengan menggunakan metode Deterministic Output Validation. Skenario pengujian dijalankan berulang sebanyak tujuh kali untuk memverifikasi stabilitas algoritma Winnowing.

Hasil pengujian pada Tabel 2 menunjukkan bahwa sistem menghasilkan skor yang identik pada setiap pengulangan untuk ketiga skenario uji. Hal ini mengindikasikan bahwa algoritma yang diterapkan bersifat deterministik dan stabil.

Tabel 2. Hasil Pengujian Konsistensi Algoritma

Skenario Pengujian	Uji 1	Uji 2	Uji 3	Uji 4	Uji 5	Uji 6	Uji 7	Status
Modifikasi Minor	100%	100%	100%	100%	100%	100%	100%	Stabil
Lintas Bahasa	0%	0%	0%	0%	0%	0%	0%	Stabil
Normalisasi Variabel	61,54%	61,54%	61,54%	61,54%	61,54%	61,54%	61,54%	Stabil

3.4. Pembahasan

Berdasarkan hasil pengujian, integrasi algoritma Winnowing dengan metode normalisasi identifier memberikan peningkatan signifikan terhadap kemampuan sistem dalam menangani plagiarisme yang disamarkan. Pada skenario perubahan nama variabel (Skenario 3), sistem tetap mampu mendeteksi kemiripan di atas 60%, yang menunjukkan bahwa pendekatan berbasis token yang dinormalisasi lebih unggul dibandingkan metode pencocokan teks biasa yang rentan gagal jika variabel diubah.

Selain itu, skor 0% pada pengujian lintas bahasa menegaskan bahwa sistem memiliki selektivitas yang baik dan bebas dari bias false positive. Stabilitas hasil yang mencapai 100% pada uji konsistensi juga menegaskan bahwa sistem ini layak digunakan sebagai alat bantu verifikasi tugas akademik yang objektif dan andal di lingkungan laboratorium informatika.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan sistem deteksi plagiarisme kode sumber berbasis web dengan mengintegrasikan algoritma Winnowing dan metode Jaccard Similarity. Sistem yang dikembangkan mampu menjamin keaslian tugas pemrograman melalui mekanisme fingerprinting yang diperkuat dengan tahap pre-processing khusus, meliputi tokenisasi dan normalisasi identifier.

Berdasarkan hasil pengujian, sistem terbukti efektif dalam mendeteksi kemiripan kode meskipun terjadi modifikasi permukaan (surface-level modification). Hal ini dibuktikan dengan tingkat deteksi sebesar 68% pada kasus modifikasi minor dan 61,54% pada kasus perubahan nama variabel secara menyeluruh. Selain itu, sistem menunjukkan reliabilitas tinggi dengan skor konsistensi 100% pada pengujian berulang dan tidak menghasilkan false positive (0%) pada pengujian lintas bahasa. Dengan demikian, sistem ini layak diterapkan sebagai solusi verifikasi tugas akademik yang objektif, efisien, dan andal di lingkungan laboratorium informatika.

5. Saran

Pengembangan selanjutnya disarankan untuk mengatasi keterbatasan performa komputasi pada perangkat dengan spesifikasi rendah dengan menerapkan teknik pemrosesan paralel atau memindahkan beban komputasi ke layanan berbasis cloud. Selain itu, cakupan deteksi dapat diperluas tidak hanya pada kemiripan leksikal dan struktural, tetapi juga pada aspek semantik dengan mengadopsi metode Abstract Syntax Tree (AST) untuk mendeteksi perubahan logika yang lebih kompleks.

Penelitian di masa depan juga diharapkan dapat mengintegrasikan sistem ini secara langsung dengan platform manajemen pembelajaran (Learning Management System) atau GitHub Classroom. Hal ini bertujuan untuk memperkaya corpus data pembandingan secara otomatis dan mempermudah proses pengumpulan tugas mahasiswa dalam skala yang lebih besar.

Referensi:

- [1] E. Ferdiana Sari dan Ekohardi, "Penerapan Github Sebagai Media E-Learning Untuk Mengetahui Keefektifan Kolaborasi Project Pada Mata Pelajaran Pemrograman Web Dan Perangkat Bergerak Di SMK Negeri 2 Surabaya," *IT-Edu*, vol. 06, no. 2, hal. 14–22, 2021.
- [2] J. Halim dan D. Lasut, "Document Plagiarism Detection Application Using Web-Based TF-IDF and Cosine Similarity Methods," *Jurnal Teknologi*, vol. 7, no. 2, 2024.
- [3] A. Soldati, A. Chiozzi, Ž. Nikolić, C. Vaccaro, dan E. Benvenuti, "A PROMETHEE Multiple-Criteria Approach to Combined Seismic and Flood Risk Assessment at the Regional Scale," *Applied Sciences*, vol. 12, no. 3, 2022.
- [4] T. Pradeep, A. K. Gonepally, A. K. Pusala, dan C. T. Singarapu, "Anti-plagiarism tool helping developers to generate authentic code," vol. 14, hal. 1208–1215, Jan. 2025.
- [5] I. G. A. Eka Putra, "Deteksi Plagiarisme Source Code Tugas Mahasiswa Menggunakan Algoritma Cosine Similarity dan Pembobotan TF-IDF," vol. 1, Nov. 2022.
- [6] M. S. Ramli, S. Cokrowibowo, dan M. F. Rustan, "Uji Plagiarism pada Tugas Mahasiswa Menggunakan Algoritma WInnowing," *Journal of Applied Computer Science and Technology*, vol. 2, no. 2, hal. 108–112, 2021.
- [7] M. E. Rijoly, W. Pramudita, B. P. Tomasouw, dan Z. A. Leleury, "Perancangan Sistem Deteksi Plagiarisme Skripsi (Judul Dan Abstrak) Berbasis Matlab Menggunakan Algoritma WInnowing," *Tensor: Pure and Applied Mathematics Journal*, vol. 2, no. 2, hal. 67–76, 2021.
- [8] I. B. K. S. Arnawa, "Implementasi Algoritma WInnowing Dalam Mendeteksi Plagiarisme Pada Tugas Mahasiswa," *Jurnal Informasi Dan Komputer*, vol. 1, 2022.
- [9] A. K. Saputra, E. Erlangga, dan T. Tanjung, "Analisis Kinerja Algoritma WInnowing pada Pendeteksian Plagiarisme," *Explore: Jurnal Sistem Informasi Dan Telematika*, vol. 13, no. 2, hal. 125, 2022.
- [10] E. Eslit, "AI-Generated Text and Plagiarism Detection: Pandora's Tech-Box Unmasked," *Preprints*, hal. 0–15, 2025.
- [11] A. Yudra Bramantya et al., "Analisis Algoritma WInnowing pada Pendeteksian Plagiarisme Judul Tugas Akhir," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, hal. 268–273, 2022.
- [12] J. Gea dan Y. A. Susetyo, "Implementasi Framework Flask Pada Modul Beta-App Pada Aplikasi Sistem Informasi Helpdesk (Sih) Studi Kasus Pt Xyz," *Jurnal Informatika*, vol. 23, no. 2, hal. 243–258, 2023.
- [13] S. Schleimer, D. S. Wilkerson, dan A. Aiken, "WInnowing: Local Algorithms for Document Fingerprinting," dalam *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2003, hal. 76–85.