DETEKSI PLAGIARISME KALIMAT BAHASA INDONESIA MENGGUNAKAN PENGEMBANGAN ALGORITMA RABIN-KARP UNTUK MENINGKATKAN EFISIENSI DAN AKURASI

SKRIPSI

DISUSUN OLEH:

WAHYU ARDIANSYAH

NPM.2109020120

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2026

DETEKSI PLAGIARISME KALIMAT BAHASA INDONESIA MENGGUNAKAN PENGEMBANGAN ALGORITMA RABIN-KARP UNTUK MENINGKATKAN EFISIENSI DAN AKURASI

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

WAHYU ARDIANSYAH
NPM.2109020120

PROGRAM STUDI TEKNOLOGI INFORMASI FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2026

LEMBAR PENGESAHAN

Judul Skripsi	:	Deteksi Plagiarisme Kalimat Bahasa Indonesia Menggunakan Pengembangan Algoritma Rabin-Karp Untuk Meningkatkan Efisiensi dan Akurasi
Nama Mahasiswa	:	WAHYU ARDIANSYAH
NPM	:	2109020120
Program Studi	:	TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing

(Dr. Zainal Azis, M.M., M.Si)
NIDN. 0113126301

Ketua Program Studi

(Fatma Sari Hutagalung, S.Kom,. M.Kom)
NIDN. 0117019301

Dekan

(Assoc. Prof. Ir. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

DETEKSI PLAGIARISME KALIMAT BAHASA INDONESIA MENGGUNAKAN PENGEMBANGAN ALGORITMA RABIN-KARP UNTUK MENINGKATKAN EFISIENSI DAN AKURASI

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing – masing disebutkan sumbernya.

Medan, 23 Mei 2026

Yang membuat pernyataan

WAHYU ARDIANSYAH

NPM.2109020120

PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, Saya bertanda tangan dibawah ini:

Nama : WAHYU ARDIANSYAH
NPM : 21090202120
Program Studi : TEKNOLOGI INFORMASI
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exlusive Royalty Free Right*) atas penelitian skripsi saya yang berjudul :

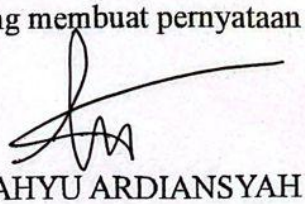
DETEKSI PLAGIARISME KALIMAT BAHASA INDONESIA MENGGUNAKAN PENGEMBANGAN ALGORITMA RABIN-KARP UNTUK EFISIENSI DAN AKURASI

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dan membentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 23 Mei 2026

Yang membuat pernyataan



WAHYU ARDIANSYAH

NPM.2109020120

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: WAHYU ARDIANSYAH
Tempat dan Tanggal Lahir	: Medan, 21 Oktober 2003
Alamat Rumah	: Jalan Cahaya, Gang Setuju NO.4
Telepon/Faks/HP	: 0813-6242-1560
E-mail	: wahyuardi1980@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	:	TAMAT : 26 Juni 2015
SMP	:	TAMAT : 28 Mei 2018
SMK	:	TAMAT : 03 Juni 2021

KATA PENGANTAR

Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Akrim, M.Pd., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Assoc. Prof. Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firahmi Rizky, S.Kom.,M.Kom Wakil Dekan 1 Fakultas Ilmu Komputer dan Teknologi Informasi (Fikti) UMSU.
4. Bapak Mhd. Basri, S.Si.,M.Kom Wakil Dekan 3 Fakultas Ilmu Komputer dan Teknologi Informasi (Fikti) UMSU.
5. Ibu Fatma Sari Hutagalung, S.Kom, M.Kom Ketua Program Studi Teknologi Informasi
6. Bapak Okvi Nugroho, S.Kom., M.Kom Sekretaris Program Studi Teknologi Informasi
7. Pembimbing saya, yaitu bapak Dr. Zainal Azis, M.M., S.Si
8. Kedua orang tua saya yang sudah mendukung saya dalam penyelesaian kuliah ini.
9. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

DETEKSI PLAGIARISME KALIMAT BAHASA INDONESIA MENGGUNAKAN PENGEMBANGAN ALGORITMA RABIN-KARP UNTUK EFISIENSI DAN AKURASI

ABSTRAK

Plagiarisme merupakan salah satu permasalahan serius dalam dunia akademik yang dapat menurunkan kualitas dan orisinalitas karya ilmiah. Kemudahan akses informasi digital menyebabkan praktik plagiarisme semakin meningkat, sehingga diperlukan sistem deteksi yang mampu bekerja secara efektif dan efisien. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi plagiarisme kalimat Bahasa Indonesia berbasis web menggunakan kombinasi algoritma Rabin-Karp dan Cosine Similarity. Algoritma Rabin-Karp digunakan untuk mendeteksi kemiripan substring secara cepat melalui teknik hashing dan k-gram, sedangkan Cosine Similarity digunakan untuk mengukur tingkat kemiripan dokumen berdasarkan representasi vektor TF-IDF. Penelitian menggunakan pendekatan kuantitatif dengan metode eksperimen terhadap dokumen berbahasa Indonesia yang terdiri dari dokumen asli dan dokumen hasil plagiasi dengan variasi copy-paste, parafrasa, serta penggunaan sinonim. Tahapan sistem meliputi preprocessing teks berupa case folding, cleaning, tokenisasi, stopword removal, dan stemming menggunakan pustaka Sastrawi. Hasil pengujian menunjukkan bahwa kombinasi kedua algoritma mampu meningkatkan efektivitas deteksi plagiarisme dibandingkan penggunaan satu metode saja. Sistem dapat mendeteksi kemiripan teks secara eksak maupun kemiripan makna dengan hasil yang lebih akurat dan efisien. Implementasi aplikasi berbasis web juga memudahkan pengguna dalam melakukan pemeriksaan dokumen secara real-time melalui antarmuka yang sederhana dan mudah diakses. Penelitian ini diharapkan dapat menjadi solusi praktis dalam mendukung integritas akademik serta memberikan kontribusi dalam pengembangan teknologi pemrosesan teks Bahasa Indonesia.

Kata Kunci : Plagiarisme, Rabin-Karp, Cosine Similarity, Deteksi Teks, Bahasa Indonesia, Web.

PLAGIARISM DETECTION OF INDONESIAN SENTENCES USING RABIN-KARP ALGORITHM DEVELOPMENT FOR EFFICIENCY AND ACCURACY

ABSTRACT

Plagiarism is a serious problem in the academic world that can reduce the quality and originality of scientific work. The ease of access to digital information has led to an increase in plagiarism practices, so that a detection system that can work effectively and efficiently is needed. This study aims to design and implement a web-based plagiarism detection system for Indonesian sentences using a combination of the Rabin-Karp and Cosine Similarity algorithms. The Rabin-Karp algorithm is used to quickly detect substring similarities through hashing and k-gram techniques, while Cosine Similarity is used to measure the level of document similarity based on TF-IDF vector representations. The study uses a quantitative approach with an experimental method on Indonesian language documents consisting of original documents and plagiarized documents with variations of copy-paste, paraphrase, and the use of synonyms. The system stages include text preprocessing in the form of case folding, cleaning, tokenization, stopword removal, and stemming using the Sastrawi library. The test results show that the combination of the two algorithms can increase the effectiveness of plagiarism detection compared to using one method alone. The system can detect exact text similarities and meaning similarities with more accurate and efficient results. The implementation of a web-based application also facilitates real-time document inspection through a simple and accessible interface. This research is expected to provide a practical solution to support academic integrity and contribute to the development of Indonesian text processing technology.

Keywords*: Plagiarism, Rabin-Karp, Cosine Similarity, Text Detection, Indonesian, Web.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah memberikan kemudahan bagi masyarakat dalam mengakses, mengelolah, dan mendistribusikan informasi, khususnya dalam bentuk teks atau kalimat digital. Kemudahan ini membawa dapak positif bagi dunia pendidikan, penelitian, dan piblikasi ilmiah karena proses pencarian referensi dan penyususan karya tulis menjadi lebih cepat dan efisien. Namun, disisi lain, kemajuan teknologi juga menimbulkan permasalahan baru, salah satunya adalah meningkatnya praktik plagiarisme.

Plagiarisme merupakan tindakan menjiplak karya orang lain, baik sebagian maupun seluruhnya, tanpa memberikan pengakuan atau atribusi yang semestinya. Dalam lingkungan akademik, plagiarisme menjadi masalah serius karena tidak hanya melanggar etika ilmiah, tetapi juga menurunkan kualitas serta orisinalitas karya ilmiah. Kemudahan akses internet dan praktik copy-paste yang tidak bertanggung jawab menyebabkan plagiarisme semakin sulit dikendalikan apabila tidak didukung oleh sistem deteksi yang memadai.

Menurut data *Turnitin Global Plagiarism Report* (2022), sekitar 38% karya tulis mahasiswa di Asia Tenggara mengandung unsur plagiarisme, dengan Indonesia menempati posisi tiga besar. Di lingkungan akademik, plagiarisme bukan hanya melanggar etika ilmiah, tetapi juga menurunkan kualitas dan orisinalitas penelitian.

Turnitin (2023), mengumumkan bahwa lebih dari 65 juta makalah telah ditinjau sejak peluncuran fitur barunya pada bulan April yang mendeteksi kemiripan dengan penulisan AI. Perusahaan juga mengumumkan bahwa dari 65 juta makalah tersebut, lebih dari 2,1 juta – 3,3 persen – telah ditandai memiliki setidaknya 80 persen tulisan AI. Hampir 6,7 juta – 10,3 persen – memiliki lebih dari 20 persen tulisan AI. Pelacakan tingkat deteksi keseluruhan menunjukkan bahwa AI generatif telah memasuki ruang kelas, namun, apakah hal ini dapat diterima atau tidak, ditentukan oleh para pendidik sendiri.

Salah satu tantangan utama dalam deteksi plagiarisme adalah karakteristik bahasa Indonesia itu sendiri, yang memiliki struktur morfologi yang kaya serta variasi sinonim yang luas. Deteksi plagiarisme tidak hanya perlu mengidentifikasi kesamaan teks atau kalimat secara eksak, tetapi juga harus mampu mendeteksi kemiripan makna akibat parafrase. Oleh karena itu, diperlukan metode yang mampu bekerja secara efektif pada level kalimat dengan mempertimbangkan kesamaan struktur dan makna.

Berdasarkan penelitian terdahulu menyatakan bahwa algortima Rabin-Karp dapat digunakan untuk mendeteksi plagiarisme dalam file dokumen dalam format teks atau kalimat. Hasil analisis yang membandingkan persentase kemiripan dari pengujian sepuluh dokumen teks dengan algoritma Rabin-Karp menghasilkan dokumen asli dan dokumen yang diuji dengan akurasi tertinggi sebesar 47.58%. Ketepatan yang paling rendah adalah 19.28%. Analisis terhadap sepuluh dokumen dengan nilai k-gram 1 menghasilkan persentase kemiripan tertinggi dan terendah masing-masing sebesar 57.14% dan 28.57%. Sedangkan nilai kemiripan 30% meliputi plagiarisme minor, plagiarisme sedang 30%-70%,

dan plagiarisme mayor lebih dari 70%. (Ari Kurniawan, S, 2023). Dalam penelitian ini Salmuasih (2013), beberapa perangkat lunak yang didesain untuk mendeteksi plagiat dokumen, diantaranya Turnitin, Eve2, CopyCatchGold, WorldCheck, Glatt, Moss, JPlag. Berdasarkan analisis informasi dari web, pendeteksi terbaik sesuai fungsinya adalah Turnitin, duplikasi dokumen dan pencocokan *string* telah banyak dibahas pada penelitian – penelitian sebelumnya. Algoritma yan digunakan diantaranya Winnowing, Smith Waterman, Boyer Moore, dan Rabin Karp namun sebagian besar tanpa menggunakan *preprocessing*, sehingga berpengaruh pada akurasi *Cosine Similarity.*

Pelaku yang melakukan plagiarisme memiliki beberapa alasan, alasan paling dominan mengapa pelaku-pelaku tindak plagiat tersebut melakukan tindakan plagiarisme adalah karena mereka malas dan merasa tindakan plagiarisme adalah sebuah jalan singkat untuk menyelesaikan tugasnya. Hal ini sering terjadi dibidang akademik dan umumnya dilakukan oleh pelajar ataupun tenaga pengajar yang ingin tugas karangan atau karya ilmiah segera selesai. Tindakan plagiarisme ini bisa berdampak kepada masyarakat berupa berkurangnya kreativitas masyarakat karena akan timbulnya rasa takut karyanya dijiplak oleh orang lain, sehingga masyarakat malas berkarya dan memunculkan ide-ide baru. (Joko Priambodo, 2018). Seiring dengan perkembangan teknologi web, kebutuhan akan sistem deteksi plagiarisme yang mudah diakses dan efisien semakin meningkat. Aplikasi berbasis web menjadi solusi yang relevan karena dapat digunakan secara luas tanpa keterbatasan platform. Melalui aplikasi web, pengguna dapat melakukan unggah kalimat, analisis kemiripan, dan memperoleh hasil deteksi secara *real-time*. Oleh karena itu, pengembangan aplikasi deteksi

plagiarisme kalimat Bahasa Indonesia berbasis web menjadi langkah strategis untuk mendukung integritas akademik. Berdasarkan uraian tersebut, penulis mengangkat topik **"Deteksi Plagiarisme Kalimat Bahasa Indonesia Menggunakan Algoritma Rabin-Karp dan Cosine Similarity"**. Penelitian ini bertujuan untuk mengkaji penerapan algoritma Rabin-Karp dalam mendeteksi kemiripan teks secara eksak dan Cosine Similarity dalam mengukur kemiripan makna, serta mengimplementasikannya dalam sebuah aplikasi berbasis web. Diharapkan penelitian ini dapat memberikan kontribusi dalam pengembangan ilmu pemrosesan teks dan bahasa alami, serta menjadi solusi praktis dalam upaya mengurangi plagiarisme pada karya ilmiah berbahasa Indonesia.

1.2 Rumusan Masalah

Berdasarkan fenomena dan latar belakang masalah yang dipaparkan di atas, maka dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Bagaimana cara kerja algoritma Rabin-Karp dalam mendeteksi kemiripan antar kalimat?
2. Bagaimana performa sistem dalam hal efisiensi waktu dan akurasi deteksi?
3. Apakah kombinasi kedua algoritma tersebut dapat meningkatkan kualitas deteksi plagiarisme?
4. Bagaimana mengimplementasikan algoritma Rabin-Karp dan Cosine Similarity ke dalam sebuah aplikasi deteksi plagiarisme berbasis web

1.3 Batasan Masalah

Supaya pembahasan masalah yang dilakukan tidak menyimpang dari pokok permasalahan, maka permasalahan yang akan dibahas dibatasi sebagai berikut:

1. Fokus pada kalimat berbahasa Indonesia.
2. Dataset berupa dokumen tugas, artikel, atau skripsi pendek.
3. Pemrosesan terbatas pada teks tertulis (tidak mencakup gambar atau tabel).
4. Sistem diimplementasikan dalam bentuk aplikasi berbasis web dengan bahasa pemrograman Python, serta memanfaatkan pustaka pendukung seperti *scikit-learn* untuk perhitungan Cosine Similarity, NLTK untuk proses tokenisasi, dan Sastrawi untuk proses *stemming* Bahasa Indonesia.

1.4 Tujuan Penelitian

Berdasarkan perumusan masalah di atas, maka dapat dideskripsikan tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang sistem deteksi plagiarisme berbasis web, yang mencakup perancangan alur proses, arsitektur sistem, serta mekanisme kerja algoritma Rabin-Karp dan Cosine Similarity.
2. Mengimplementasi algoritma Rabin-Karp untuk deteksi *substring* kemiripan dalam teks atau kalimat dan mengimplementasikan Cosine Similarity berbasi representasi vector dokumen.
3. Menguji dan menganalisis tingkat akurasi serta efektivitas sistem dalam mendeteksi plagiarisme.

1.5 Manfaat Penelitian

Berdasarkan latar belakang di atas, maka dapat dideskripsikan manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan solusi teknologi berbasis web yang praktis untuk institusi pendidikan dalam mendeteksi plagiarisme.

2. Menyediakan metode ringan dan efisien yang dapat diterapkan pada sistem pemeriksa tugas, laporan, atau skripsi.
3. Menambah literatur akademik mengenai pemrosesan teks Bahasa Indonesia.

BAB II

LANDASAN TEORI

2.1 Plagiarisme Teks

Plagiarisme merupakan tindakan menjiplak atau menyalin karya orang lain tanpa mencantumkan sumber yang jelas. Dalam konteks akademik, plagiarisme teks sering terjadi pada skripsi, jurnal, maupun artikel ilmiah. Menurut Indonesian Higher Education Law No. 20 Tahun 2003, plagiarisme dapat dikenakan sanksi akademik yang serius.

Plagiarisme adalah praktik penyalahgunaan hak kekayaan intelektual milik orang lain orang dan pekerjaan itu diakui tidak sah sebagai akibat dari pekerjaan pribadi. Studi empiris yang dilakukan oleh Hutton and French in Hartanto mengemukakan bahwa bahwa faktor-faktor yang menyebabkan plagiarisme adalah kemalasan mereka sendiri, karena mereka merasa stres, memiliki keyakinan bahwa perilakunya tidak akan diketahui, dan perilakunya bukanlah hal yang salah untuk dilakukan atau berbahaya. Filcha, A & Hayaty, M. (2019).

Namun, Mulyana menyatakan bahwa cara mencantumkan relevansi mengarahkan mahasiswa untuk melakukan duplikasi atau plagiarisme. Sadar atau tidak, cara mengutip yang dilakukan telah mendekatkan skripsi mereka pada skripsi orang lain. Dari sinilah antara lain gejala plagiarisme muncul (Mulyana, 2010)

2.1.1. Bentuk Plagiarisme

Adapun jenis-jenis plagiarisme yang diukur mulai dari yang jarang sampai yang sering terjadi dan dari yang ringan sampai yang paling parah, yaitu:

1. ***Secondary source*** **(sumber sekunder)**: Plagiasi tipe ini dimungkinkan terjadi ketika peneliti memanfaatkan sumber-sumber sekunder (seperti *literature review*). Peneliti hanya mengutip sumber - sumber primer yang disebut dalam sumber sekunder yang dibacanya dan tidak memberikan informasi (mengutip) sumber sekunder yang dibacanya.
2. ***Invalid Source*** **(Sumber tidak valid)**: Plagiasi jenis ini terjadi ketika peneliti memberikan informasi yang salah atau tidak memadai terhadap sumber-sumber referensi yang digunakannya.
3. ***Duplication*** **(Duplikasi)**: Plagiasi ini terjadi ketika peneliti menggunakan karya ilmiahnya sebelumnya tanpa memberikan informasi bahwa itu merupakan penelitian yang sudah dilakukan sebelumnya.
4. ***Paraphrasing*** **(Parafrase)**: Plagiasi jenis ini berupa mengambil teks dari suatu sumber, kemudian dilakukan parafrasa namun tidak disebut sumbernya, seakan teks tersebut asli miliknya
5. ***Repetitive Research*** **(Penelitian Berulang)**: Plagiasi ini ketika peneliti menggunakan data dan metode yang sama untuk penelitian tanpa menyebutkan bahwa metode itu pernah digunakan pada penelitian sebelumnya.
6. ***Replication*** **(Replikasi)**: Plagiasi ini berupa tindakan mengirimkan naskah ke beberapa saluran publikasi (jurnal, konferensi, dan lain-lain).

7. ***Misleading Attribution*** **(Atribusi yang sesat)**: salah atau tidak memadai dalam penyebutan pihak pihak yang terlibat dan berkontribusi dalam sebuah penelitian.
8. ***Unethical Collaboration*** **(kolaborasi tidak etis)**: Plagiasi jenis ini bisa terjadi ketika orang-orang yang berkolaborasi melanggar kesepakatan dan etika kolaborasi.
9. ***Verbatim Plagiarism*** **(Plagiasi kata demi kata)**: Plagiasi ini berupa tindakan mengcopy kata perkata ide atau karya orang lain tanpa menambahkan kutipan atau rujukan.
10. ***Complete Plagiarism*** **(Plagiasi total):** Tindakan plagiasi yang dilakukan penulis dengan cara menjiplak atau mencuri hasil karya orang lain seluruhnya dan mengklaim sebagai karyanya.

Bila dilihat dari berbagai macam bentuk-bentuk praktek plagiarisme di atas, dapat disimpulkan bahwa tindakan plagiarisme yang terjadi di dunia akademis lebih cenderung kepada tindakan menggunakan kembali suatu bagian dokumen teks. Kalimat/ kata dari suatu sumber yang tidak mengikuti tata aturan hak cipta, seperti aturan pengutipan maupun ketidakjelasan sumber/ pengarang asli (Purwitasari et al., 2010).

2.1.2. Dampak Plagiarisme

Dalam penulisan artikel ilmiah maka plagiarisme bisa menyebabkan dampak serius, mulai dari kehilangan kredibilitas sampai sanksi hukum dan akademik. Plagiarisme bisa merusak integritas ilmiah, menghambat inovasi, dan merugikan kredibilitas penulis serta lembaga terkait. Pada beberapa kondisi, kegiatan plagiat dilakukan demi kepraktisan karena tinggal menjiplak karya

ilmiah orang lain. Hanya saja tindakan ini tentu salah, karena sudah merugikan orang lain dengan mengakui hasil kerja keras dan buah pikirannya sebagai hasil kerja keras diri sendiri.

Proses mengambil atau menjiplak karya ilmiah orang lain pada dasarnya diperbolehkan. Hanya saja ada aturannya, yaitu mencantumkan kredit, sitasi, atau sumber setelah maupun sebelum kalimat tersebut dimasukan ke dalam karya ilmiah yang sedang disusun. Maka karya yang ditulis bebas plagiat sekalipun tetap mengambil beberapa materi yang setelah itu ditulis dalam bentuk kutipan. (Gusnayetti, 2025).

Plagiasi mempunyai dampak yang luas, baik bagi individu yang melakukannya maupun bagi institusi yang terkait. Adapun beberapa dampak utama dapat dilihat pada uraian sebagai berikut:

1. **Konsekuensi Akademik**

   Plagiasi bias mengakibatkan sanksi akademik seperti pencabutan gelar, pembatalan publikasi, atau bahkan skorsing dari institusi pendidikan. Banyak universitas memiliki kebijakan tegas terhadap plagiasi dan dapat memberikan hukuman berat bagi pelanggar. Kebijakan ini biasanya mencangkup penggunaan perangkat lunak deteksi plagiasi seperti Turnitin atau Grammarly sebagai preventif.

2. **Kerugian Reputasi**

   Bagi peneliti atau akademisi, plagiasi bias menghancurkan reputasi profesional mereka. Kredibilitas yang dibangun selama bertahun-tahun bisa hancur dalam sekejap akibat temuan plagiasi pada publikasi ilmiah mereka. Reputasi yang buruk ini dapat juga berdampak negatif

pada kesempatan mendapatkan pendanaan penelitian atau kolaborasi akademik di masa depan.

3. **Implikasi Hukum**

   Dari banyaknya kasus plagiasi bisa berujung pada tuntutan hukum, terutama jika karya yang dikutip merupakan hak cipta yang dilindungi. Institusi akademik dan penerbit jurnal biasanya mempunyai kebijakan ketat tentanghak cipta dan etika penulisan. Beberapa negara bahkan mempunyai regulasi ketat terhadap plagiasi dalam karya akademik.

4. **Menurunkan Kualitas Penelitian**

   Plagiasi bisa berdampak pada mutu penelitian secara keseluruhan. Disaat seseorang hanya menyusun kembali karya orang lain tanpa kontribusi baru, ilmu pengetahuan tidak berkembang dengan baik. Keaslian penelitian sangat penting dalam memastikan adanya perkembangan dan inovasi dalam berbagai bidang keilmuan.

Dalam menjauhkan plagiasi, ada langkah-langkah yang bisa diambil oleh para akademisi dan penulis artikel ilmiah. Adapun strategi yang efektif yang akan diambil tersebut adalah sebagai berikut:

1. **Memahami Aturan dan Etika Penulisan Akademik**

   Penulis harus memahami etika akademik dalam penulisan ilmiah, termasuk bagaimana cara mengutip sumber dengan benar sesuai dengan gaya kutipan yang digunakan (APA, MLA, Chicago, dll.). Memahami dasar-dasar etika akademik juga membantu menghindari kesalahan yang dapat berujung pada plagiasi yang tidak disengaja.

2. **Menggunakan Parafrase dengan Benar**

Parafrase merupakan salah satu cara untuk menghindari plagiasi, tetapi harus dilakukan dengan hati-hati. Pastikan bahwa ide atau informasi yang diambil diungkapkan dengan kalimat yang benar-benar berbeda, serta tetap mencantumkan sumber aslinya. Teknik parafrase yang baik melibatkan pemahaman mendalam terhadap materi sebelum menyusun ulang dengan gaya bahasa sendiri.

3. **Mencantumkan Sumber dengan Tepat**

   Setiap informasi yang bukan merupakan hasil pemikiran sendiri harus diberikan atribusi yang jelas. Gunakan kutipan langsung atau kutipan tidak langsung sesuai dengan standar akademik yang berlaku. Hal ini dapat dilakukan dengan menggunakan perangkat lunak manajemen referensi seperti Mendeley, Zotero, atau EndNote untuk memastikan akurasi dalam penyusunan daftar pustaka.

2.2 Representasi Teks dalam Komputasi

Dalam bidang komputasi, terutama pada pemrosesan bahasa alami (*Natural Language Processing*/NLP), representasi teks adalah proses mengubah teks yang berbentuk bahasa manusia (*natural language*) menjadi bentuk yang dapat dipahami dan diolah oleh komputer. Representasi ini sangat penting dalam berbagai aplikasi, salah satunya adalah deteksi plagiarisme teks, di mana sistem harus membandingkan dua atau lebih dokumen untuk menilai tingkat kesamaan isinya.

Dalam proses tersebut, tahapan paling fundamental adalah representasi teks, yang mengubah kalimat menjadi bentuk numerik agar bisa dianalisis secara

komputasional. Apabila proses representasi ini tidak mampu menangkap makna semantik dari teks dengan baik, maka hasil penilaiannya bisa jauh menyimpang dari penilaian manusiawi (Maulidya Prastita Syah et al., 2025).

2.2.1 Preprocessing Teks

Text preprocessing adalah proses mengubah data tekstual yang tidak terstruktur menjadi data terstruktur untuk disimpan dalam database (Arsad, A et al., 2024). Seperangkat indeks istilah yang dapat mewakili dokumen adalah tujuan dari preprocessing. Ada beberapa bagian pada bagian preprocessing teks, antara lain:

1. Tokenisasi

   Tokenisasi adalah sebuah proses untuk memilah isi teks sehingga menjadi satuan kata-kata. Proses ini cukup rumit untuk sebuah program komputer karena beberapa karakter dapat dijadikan sebagai pembatas (*delimiter*) dari token-token itu sendiri. (Setiawan, A et al., 2015)

2. Filtering

   Filtering merupakan proses dalam *text preprocessing* setelah tokenisasi, filtering dilakukan untuk mengambil kata penting hasil tokenisasi. Proses filtering dalam membuang kata-kata yang tidak digunakan atau *stopword* terdapat dalam *bag of words*. (Yuniar, E et al., 2022)

3. Stemming

   Stemming merupakan tahapan proses lanjutan setelah *filtering* yang digunakan untuk membuang imbuhan awalan atau akhiran menjadi kata dasar. *Library* yang digunakan pada program aplikasi ini adalah *stemming*. (Setiawan, A et al., 2015)

2.2.2 Kelebihan Preprocessing Teks

Preprocessing teks memainkan peran sentral dalam sistem deteksi plagiarisme untuk bahasa Indonesia karena langkah-langkah seperti case-folding, pembersihan tanda baca, tokenisasi, penghilangan stopword, dan stemming/normalisasi secara konsisten mengurangi variasi permukaan pada teks sehingga teknik berbasis perbandingan string atau fingerprint (mis. winnowing) dapat menemukan kecocokan yang semula tersembunyi oleh imbuhan atau perbedaan kapitalisasi. Berikut kelebihan dari preprocessing teks:

1. Mengurangi variasi bentuk kata sehingga perbandingan menjadi lebih akurat
    - Stemming menyamakan variasi morfologis (“membaca”, “baca”) sehingga algoritma fingerprinting atau n-gram menemukan kecocokan yang sebenarnya tersembunyi, stemming dapat meningkatkan akurasi deteksi pada studi bahasa indonesia.
2. Mengurangi ukuran dan menghemat waktu pemrosesan
    - Penghapusan *stopwords* dan tanda baca membuat jumlah token yang diproses lebih sedikit sehingga komputasi (*hashing*, *winnowing*, perhitungan similarity) lebih cepat dan memori lebih hemat serta penting digunakan saat membandingkan dokumen besar atau banyak dokumen. Beberapa implementasi aplikasi deteksi plagiat Bahasa Indonesia menggunakan Sastrawi untuk *stopword* atau *stemming* demi efisiensi.
3. Meningkatkan ketahanan terhadap variasi format dalam dokumen

- *Case-folding*, normalisasi *whitespace*, dan penghilangan markup (HTML, PDF *artifacts*) membuat sistem tahan terhadap perbedaan f*ormatting* (*copy-paste* dari web, PDF teks) sehingga fokus pada isi teks. Studi tinjauan menekankan perlunya pra-proses untuk standar input sebelum ekstraksi fitur.

2.2.3 Kekurangan Preprocessing Teks

preprocessing yang agresif dapat mengaburkan jejak plagiarisme berbentuk parafrase. penghapusan *stopword* dan *stemming* yang terlalu kuat bisa menghilangkan informasi stilistik dan struktur kalimat yang berguna untuk membedakan antara kutipan yang sah, parafrase yang wajar, dan parafrase yang dimaksudkan untuk menyamarkan salinan. Berikut kekurangan dari *preprocessing teks*:

1. Berpotensi menghilangkan jejak plagiarisme paraphrase
    - Penghapusan *stopwords* dan *stemming* berlebihan dapat menghapus pola gaya penulisan yang berguna untuk mendeteksi parafrase—mis. jika pelaku mengubah struktur kalimat tapi mempertahankan gagasan, beberapa teknik *preprocessing* bisa mengaburkan bukti itu sehingga deteksi paraphrase menjadi lebih sulit. Studi perbandingan menunjukkan *trade-off* antara pembersihan dan kemampuan menangkap parafrase.
2. Kesalahan *stemming* atau normalisasi untuk bahasa indonesia
    - *Stemming* yang tidak sempurna (atau yang tidak cocok untuk variasi penulisan) dapat menghasilkan *over-stemming*

(menggabungkan kata yang berbeda secara makna) atau *under-stemming*, sehingga menurunkan presisi. Beberapa studi lokal (tesis atau jurnal) menemukan hasil yang bervariasi tergantung implementasi.

3. Menghapus informasi semantik penting
    - *Stopword removal* atau penghapusan kata bantu bisa menghilangkan kata-kata yang, walau sering, membawa konteks penting untuk membedakan antara kutipan yang benar dan plagiarisme kontekstual (mis. perbedaan antara klaim dan kutipan). Oleh sebab itu beberapa sistem memilih untuk tidak menghapus semua *stopwords* atau memperlakukan kata-kata tertentu secara khusus.

2.2.4 Model Representasi Teks

Model representasi teks adalah cara mengubah teks bahasa alami menjadi bentuk numerik atau simbolik agar dapat diproses komputer. Representasi ini penting untuk tugas komputasi seperti pencarian informasi, klasifikasi teks, dan deteksi plagiarisme. Adapun dua mode umum yang digunakan dalam deteksi plagiarisme adalah:

1. *Vector Space Model (VSM)*

    Model ruang vektor adalah model sistem temu balik informasi yang mengibaratkan masing-masing *query* dan dokumen sebagai sebuah vektor N-dimensi. Tiap dimensi pada vektor tersebut diwakili oleh satu *term*. *Term* yang digunakan biasanya berpatokan kepada *term* yang ada pada

query, sehingga *term* yang ada pada dokumen tetapi tidak ada pada *query* biasanya diabaikan. (Alun & Anggun, 2021)

2. *Bag Of Words (BoW)*

   Bag-of Words merupakan sebuah model dari sebuah proses yang ada didalam *Natural Language Processing(NLP)*, dan banyak digunakan untuk mengambil nilai dari sebuah kata yang sebelumnya diolah pada sebuah model machine learning. Model *Bag-of-Words* bekerja dengan cara mempelajari sebuah kata dari pada sebuah dokumen, kemudian menginterpretasikan setiap dokumen dengan menghitung jumlah kemunculan tiap kata dari dokumen tersebut. (Raja Farhan, R et al., 2022)

2.2.5 Kelebihan Model Representasi Teks

Salah satu kelebihan dengan penggunaan representasi yang lebih kaya (misalnya *embedding* atau model semantik) adalah mampu menangkap kemiripan makna yang tidak hanya secara literal (misal kata yang sama), tapi juga yang terparafrasa atau menggunakan sinonim. Contohnya dalam penelitian "Pengukuran Kemiripan Kalimat Bahasa Indonesia Menggunakan *Representasi* Word *Embedding FastText*", yang memakai rata-rata vektor kata dengan model *FastText* pralatih, menghasilkan korelasi yang baik terhadap skor kemiripan manusia (*semantic textual similarity*). Berikut kelebihan dari model representasi teks:

1. VSM atau TF-IDF (Bag of Words)
   - TF–IDF dan model vektor sederhana masih sering dipakai untuk deteksi plagiarisme karena implementasinya ringan, cepat dihitung

untuk korpus besar, dan hasilnya mudah diinterpretasikan (mis. skor cosinus antar dokumen).

2. N-gram atrau fingerprinting (char atau word n-gram)
    - Representasi berbasis n-gram (terutama *character* n-gram) memperbaiki masalah perubahan kecil (mis. Penyisipan atau penyuntingan) karena menangkap pola *substring* dan robus terhadap perubahan kata. *Fingerprinting* (*hashing substrings*) efisien untuk pencocokan cepat antar dokumen besar.
3. Pendekatan hibrid dan praktik terbaik
    - Banyak *paper* dan aplikasi nyata menunjukkan pendekatan terbaik adalah hibrida: gunakan *fingerprinting* atau n-gram untuk menangkap potongan identik & perubahan kecil, TF-IDF untuk *baseline* cepat, dan *embedding* berbasis *transformer* atau *sentence encoder* untuk menangkap parafrase atau terjemahan. Sistem produksi sering memakai tahap *retrieval* cepat (TF-IDF atau *Faiss with shallow embeddings*) untuk mereduksi kandidat, lalu lakukan *scoring* mendalam (SBERT/BERT + *alignment*) pada kandidat top-k.

2.2.6 Kekurangan Model Representasi Teks

Salah satu kekurangan model representasi berbasis *embedding* adalah bahwa dalam kasus plagiarisme ekstrem atau parafrase yang sangat bebas, meskipun *embedding* menangkap aspek semantik, bisa saja masih gagal mendeteksi bahwa satu teks disalin secara ide atau gagasan terutama jika struktur, frase, konteks secara keseluruhan sangat diubah. *Embedding* bisa “meredam”

perbedaan penting karena *averaging* vektor kata dapat menghilangkan urutan kata, bobot penting kata, atau konteks lokal. Berikut kekurangan model representasi teks:

1. *Word embeddings* tradisional (Word2Vec, Doc2Vec, FastText)
    - Model kata per kata (Word2Vec) tidak langsung menangkap konteks kalimat atau urutan panjang kamu masih membutuhkan *pooling* atau *aggregation* untuk level dokumen Doc2Vec memerlukan dokumen latih yang bagus dan performa menurun bila plagiarisme sangat tersetruktur ulang (ulang susunan kalimat). Studi khusus Bahasa Indonesia menunjukkan Word2Vec atau Doc2Vec efektif untuk similarity dibanding TF–IDF pada beberapa kasus.
2. *Setence* atau dokument *encoers* dan *retrieval* (*USE*, *SBERT*, *siamese nets*)
    - Butuh sumber daya komputasi (GPU untuk *fine-tune*), dan *fine-tuning* pada Bahasa Indonesia atau *domain* akademik sering diperlukan karena model pra-latih umumnya didominasi data bahasa Inggris. Ada penelitian yang menerapkan USE atau BERT + Faiss untuk plagiarisme atau penugasan dengan hasil baik.
3. *Transformer* atau BERT dan model bahasa khusus (indoBERT / XLM-R)
    - Biaya komputasi dan kebutuhan dataset latih yang besar, serta *latency* lebih tinggi untuk sistem *real-time*, model juga rentan terhadap *adversarial rewriting* yang memakai sinonimi jarang atau struktur kalimat yang sangat berbeda. Beberapa studi perbandingan

melaporkan BERT atau *transformer* mengungguli Word2Vec atau *FastText* dan TF-IDF pada tugas *similarity*.

2.3 Rabin-Karp

Algoritma Rabin-Karp adalah salah satu algoritma pencocokan string (*string matching)* yang diperkenalkan oleh Richard M. Karp dan Michael O. Rabin pada tahun 1987. Tujuannya adalah mencari keberadaan suatu pola (*pattern*) dalam sebuah teks (*text*) dengan cara yang efisien.

Berbeda dengan metode pencocokan sederhana (*brute-force*) yang membandingkan pola dengan teks karakter demi karakter, Rabin-Karp menggunakan teknik *hashing* untuk mempercepat proses pencarian. Teknik ini memungkinkan pencocokan dilakukan dengan lebih cepat, terutama jika pola yang dicari muncul berkali-kali dalam teks.

Algoritma Rabin-Karp memiliki beberapa karakteristik yaitu menggunakan K-Gram dan *hashing*. Penerapan algoritma Rabin-Karp dilkakukan setelah melewati tahapan *preprocessing*.(Filcha, A., & Hayaty, M) Berikut tahapan algoritma Rabin-Karp.

1. *K-Gram*. K-gram adalah rangkain token yang panjang dengan panjang k. Metode K-Gram ini mengambil potongan - potongan karakter huruf sejumlah nilai k dari sebuah teks yang secara kontiunitas dibaca dari awal teks sumber hingga akhir teks sumber. (Filcha, A., & Hayaty, M) Contoh K-Gram dengan nilai k = 4 dapat dilihat pada Tabel I.

TABEL I
CONTOH K-GRAM

Kalimat	Komputer adalah perangkat elektronik
Preprocessing	komputerperangkatelektronik
K-Gram{4}	{komp}{ompu}{mput}{pute}{uter} {terp}{erpe}{rper}{pera}{eran} {rang}{angk}{ngka}{gkat}{kate} {atel}{tele}{elek}{lekt}{ektr} {ktro}{tron}{roni}{onik}

Tabel 2.1 K-gram (Filcha, A., & Hayaty, M)

2. *Hashing*. Hashing adalah merupakan salah satu cara untuk mengubah karakter *string* menjadi *integer* yang disebut nilai *hash*. Proses pengubahan menjadi nilai *hash* menggunakan fungsi *rolling hash*. (Filcha, A., & Hayaty, M) persamaan *rolling hash* dapat dilihat pada persamaan I.

$$H(c_1 \cdots c_k) = \left(c_1 \cdot b^{\{k-1\}} + c_2 \cdot b^{\{k-2\}} + \cdots + c_{\{k-1\}} \cdot b^1 + c_k\right) mod\ q$$

Persamaan I Rabin-Karp (Filcha, A., & Hayaty, M)

Keterangan :

H : substring

C : nilai ascii per-karakter

B: konstan bilangan prima

K: banyak karakter

Q: modulo bilangan prima

Berikut contoh perhintungan *rolling hash* terhadap *substring* maka dengan nilai K-Gram 4 dapat dilihat pada Tabel II.

TABEL II CONTOH *ROLLING HASH*	
Attribut	**Nilai Array**
Rolling Hash Pertama	[0] => maka m=109, a=97, k=107, a=97, basis=11, mod= 10007 H=c_m*b^(k-1)+c_a*b^(k-2)+c_k*b^(k-3) +c_a*b^(k4) H=109*11^3+97*11^2+107*11^1+97*11^0H=145079 +11737+1177+97 H=158090 Mod 10007 H=7985
Rolling Hash Kedua	[1] => akan a=97, k=107, a=97, n=110, basis=11, mod= 10007 H=c_a*b^(k-1)+c_k*b^(k-2)+c_a*b^(k-3) +c_n*b^(k4) H=97*11^3+107*11^2+97*11^1+110*11^0 H=129107+12947+1067+110 H=143231 Mod 10007 H= 3133

Tabel 2.2 Rolling Hash *(Filcha, A., & Hayaty, M)*

2.3.1 Kelebihan Rabin-Karp

Kelebihan Rabin-Karp dalam memeriksa dokumen dalam jumlah yang besar, sehingga sangat cocok untuk digunakan dalam sistem pendeteksi plagiarisme pada skripsi atau tugas akademik lainnya, metode ini memungkinkan pengajar untuk mengevaluasi keaslian dokumen yang diserahkan mahasiswa, berikut kelebihan dari Rabin-Karp:

1. Efisiensi untuk pencarian *Exact Match* (*Copy-Paste*)

a. Rabin-Karp sangat baik dalam mendeteksi *plagiarisme* yang bersifat langsung (*Copy-Paste*), dimana kalimat atau paragraf disalin tanpa perubahan.
b. Dengan *rolling hash*, proses pencarian *substring* dilakukan lebih cepat dibandingkan pencarian karakter demi karakter (*brute force*).

2. Mendukung Pencarian Banyak Pola
    a. Rabin-Karp mampu mencari beberapa pola sekaligus dalam bentuk teks, cukup dengan menghitung nilai *hash* untuk setiap pola, hal ini berguna dalam mendeteksi *plagiarisme* antara banyak dokumen (*multi-dokumen*).
3. Skalabilitas untuk Teks Panjang
    a. Untuk teks yang panjang (misalnya artikel, skripsi, atau laporan), Rabin-Karp tetap relatif cepat karena memanfaatkan *rolling hash*.

2.3.2 Kekurangan Rabin Karp

Kekurangan Rabin-Karp sulitnya keakuratan antar kata yang mirip, karena pada algoritma Rabin-Karp masih menggunakan fungsi *hash* untuk mengubah kata menjadi sebuah bilangan desimal, berikut kekurangan dari Rabin-Karp:

b. Rentan terhadap *Collision Hash*

    Collision terjadi ketika substring berbeda memiliki nilai hash yang sama. Hal ini memaksa algoritma melakukan pengecekan karakter manual atau menurunkan efisiensi. Jika dokumen sangat besar, collision bisa sering terjadi dan memperlambat kinerja.

c. Sensitif terhadap Perubahan Kecil

Jika ada perubahan kecil pada teks (misalnya menambahkan tanda baca atau mengubah satu huruf), maka nilai *hash substring* berubah total. Akibatnya Rabin-Karp tidak akan mengenali teks sebagai sama, meskipun secara semantik maknanya identik.

d. Kurang Memahami Semantik

Rabin-Karp hanya bekerja pada level sintaksis (karakter dan *string*). Tidak ada kemampuan untuk memahami makna (semantik) teks. Oleh karena itu, sulit digunakan untuk mendeteksi *plagiarisme* yang melibatkan modifikasi bahasa, sinonim, atau parafrasa.

2.4 Cosine Similarity

Cosine Similarity mengukur kemiripan antara dua dokumen atau teks. Pada Cosine Similarity dokumen atau teks dianggap sebagai vector. Untuk pencocokan teks, nilai dari vector A dan B adalah vector term-frequency dari dokumen. Nilai Cosine Similarity berada pada range 0-1 (Ardi, S et al., 2023). Persamaan Cosine Similarity disajikan pada rumus persamaan I sebagai berikut :

$$(dj, q) = \frac{\sum_{i=l}^{t}(wij.wiq)}{\sqrt{\sum_{i=l}^{t} w_{ij.}^{2} \sum_{i=l}^{t} w_{iq}^{2}}}$$

Rumus persamaan I Cosine Similarity (Filcha, A., & Hayaty, M)

2.4.1 Kelebihan Cosine Similarity

Cosine Similarity adalah metode sederhana, efisien, dan sangat populer untuk mengukur kemiripan antar teks. Kelebihannya adalah kemudahan implementasi, ketahanan terhadap perbedaan panjang dokumen, serta efektivitas

pada teks yang memiliki kesamaan kata, berikut beberapa kelebihan dari cosine similarity:

a. Tidak Terpengaruh Panjang Dokumen

   Cosine Similarity tidak melihat panjang teks, melainkan arah vektornya, dua dokumen dengan panjang berbeda tetapi isinya mirip tetap bisa dikenali sebagai mirip. Contohnya dokumen A (100 kata) dan dokumen B (200 kata) bisa tetap terdeteksi mirip jika menggunakan kata – kata serupa.

b. Mudah Dihitung dan Diimplementasikan

   Formula Cosine Similarity sederhana, hanya melibatkan *operasi dot product* dan norma *vector*. Sangat cocok untum implementasi praktis dalam sistem deteksi plagiarisme.

c. Efisiensi untuk Big Data

   Dapat digunakan pada dataset besar dengan bantuan optimasi seperti *sparse matrix*. Banyak dipakai dalam mesin pencari, sistem rekomendasi, dan deteksi plagiarisme skala besar.

2.4.2 Kekurangan Cosine Similarity

Salah satu kekurangan utama dari Cosine Similarity adalah ketebatasannya dalam memahami makna atau semantik dari sebuat kata. Pendekatan ini hanya berfokus pada representasi numerik dari kata – kata, biasanya dalam bentuk *term frequency* atau *TF-IDF*, tanpa mempertimbangkan hubungan makna antar kata, berikut beberapa kekurangan cosine similarity:

a. Sensitif terhadap Representasi Teks

Hasil Cosine Similarity sangat bergantung pada cara teks direpresentasikan. Dengan BoW, kata umum (misalnya “dan”, “yang”) bisa menggangu hasil jika tidak dihapus (*stopword removal*). Dengan *TF-IDF*, kata – kata jarang lebih menonjol, tetapi tidak menyelesaikan masalah sinonim/parafrasa.

b. Tidak Bisa Membedakan Tingkat Plagiarisme dengan Detail

Cosine Similarity hanya memberi skor antara 0-1. Tidak bisa secara langsung menunjukkan bagian mana dari dokumen yang mirip. Untuk analisis plagiarisme yang lebih detail, perlu algoritma tambahan seperti Rabin-Karp atau metode *substring matching*.

c. Beban Komputasi untuk dokumen Besar

Jika dataset berisi ribuan atau jutaan dokumen, menghitung Cosine Similarity antar semua pasangan dokumen bisa menjadi sangat mahal secara komputasi.

2.5 Faktor – Faktor yang Mempengaruhi Rabin Karp

a. **Panjang *Substring***: Semakin panjang *substring* yang dibandingkan, semakin kecil kemungkinan terjadi *collision* (tabrakan *hash*), tapi juga semakin besar waktu komputasi. Jika terlalu pendek banyak *false positive* (kesamaan palsu), dan jika terlalu panjang bisa melewatkan plagiarisme yang dimodifikasi sedikit.

b. **Fungsi *Hash***: yang dipilih mempengaruhi kecepatan dan akurasi. Fungsi *hash* sederhana (misalnya *rolling hash*) lebih cepat, tetapi lebih rentan terhadap *collision*. *Hash* yang kompleks (seperti polinomial *rolling hash*) mengurangi *collision* tapi menambah waktu proses.

c. **Ukuran *Prime Number* atau Modulus**: Pemilihan bilangan prima untuk modulus dalam perhitungan *hash* mempengaruhi seberapa unik hasil *hash*-nya. Jika modulus terlalu kecil menyebabkan banyak tabrakan hash, dan jika modulus terlalu besar memakan waktu perhitungan yang lama.

d. ***Preprocessing* dan Normalisasi Teks**: Penghapusan tanda baca, huruf kapital, dan *stopwords* dapat meningkatkan hasil deteksi plagiarisme. Teks yang tidak dinormalisasi bisa membuat Rabin–Karp mendeteksi perbedaan kecil sebagai berbeda total.

e. **Panjang Dokumen dan Jumlah Pola**: Semakin besar dataset atau jumlah dokumen yang dibandingkan, semakin tinggi kompleksitas waktunya.

2.6 Faktor – Faktor yang Mempengaruhi Cosine Similarity

a. **Representasi Teks (Model Vektor)**: Hasil sangat bergantung pada bagaimana teks diubah menjadi vektor seperti *Bag-of-Words* (BoW): hanya menghitung frekuensi kata, TF-IDF menyesuaikan bobot kata berdasarkan kepentingan, Word2Vec atau BERT mempertimbangkan makna semantik.

b. **Pra-pemrosesan Teks**: Penghapusan *stopwords*, *stemming*, dan *lemmatization* dapat meningkatkan akurasi dengan mengurangi kata tidak penting. Contohnya “membaca” dan “dibaca” akan dianggap lebih mirip setelah *stemming*.

c. **Panjang Dokumen**: Cosine similarity relatif tidak dipengaruhi panjang dokumen (karena dinormalisasi), tapi distribusi kata tetap berpengaruh.

Dokumen panjang dengan banyak kata berbeda bisa memiliki nilai kemiripan lebih rendah meski berisi ide serupa.

d. **Kualitas Tokenisasi**: Jika tokenisasi (pemisahan kata) salah, vektor jadi tidak representatif dan hasil similarity menurun.
e. **Threshold Kemiripan**: Penentuan ambang batas (misalnya 0.7 = dianggap mirip) sangat penting. Threshold terlalu rendah maka banyak *false positive*, dan jika *threshold* terlalu tinggi maka *false negative* (plagiarisme terlewat).

2.7 Aplikasi Deteksi Plagiarisme Berbasis Web

Aplikasi deteksi plagiarisme berbasis web merupakan salah satu bentuk sistem informasi yang dirancang untuk mengidentifikasi tingkat kemiripan teks atau dokumen secara otomatis melalui jaringan internet. Aplikasi semacam ini memberikan kemudahan akses kepada pengguna karena tidak memerlukan instalasi perangkat lunak khusus di komputer pribadi, pengguna cukup menggunakan browser untuk mengirimkan dokumen atau teks yang ingin dicek kemiripannya. Konsep ini sangat relevan dalam lingkungan akademik karena memudahkan civitas akademik dalam melakukan pemeriksaan plagiarisme terhadap tugas, laporan, maupun skripsi. Sebagai contoh, penelitian Herianto., dan Yulisman, dkk (2021). menyatakan bahwa aplikasi deteksi plagiarisme berbasis web dengan algoritma Rabin-Karp dapat mempermudah proses pemeriksaan kemiripan dokumen tugas akhir mahasiswa melalui aplikasi web cek plagiarisme sehingga menjadi lebih efisien dibandingkan pemeriksaan manual.

Pendekatan aplikasi berbasis web umumnya menggunakan arsitektur client–server, di mana pengguna berinteraksi melalui antarmuka web (client) dan server bertugas memproses data yang dikirimkan, termasuk melakukan *preprocessing* teks serta menghitung tingkat kemiripan. Arsitektur ini memungkinkan pemrosesan dilakukan di satu tempat terpusat, sehingga standar pemeriksaan dan pengelolaan data menjadi lebih konsisten daripada pemeriksaan lokal di setiap perangkat. Pendekatan semacam ini juga ditemukan dalam penelitian serupa yang mengembangkan sistem pendeteksian plagiarisme dokumen berbasis web guna menghasilkan output persentase kemiripan secara otomatis, Hardison dan Maulana Ardhiansyah (2023).

Dalam implementasinya, aplikasi deteksi plagiarisme berbasis web umumnya mengintegrasikan tahapan pemrosesan teks (*text preprocessing*), representasi teks, dan perhitungan tingkat kemiripan menggunakan metode tertentu seperti algoritma Rabin-Karp dan Cosine Similarity. Hasil perhitungan kemiripan kemudian disajikan dalam bentuk persentase kemiripan kepada pengguna sebagai indikator tingkat plagiarisme sehingga dapat membantu pengguna dalam pengambilan keputusan terkait keaslian karya, Herianto., dan Yulisman, dkk (2021).

Pada penelitian ini, aplikasi berbasis web digunakan sebagai sarana implementasi algoritma Rabin-Karp dan Cosine Similarity untuk mendeteksi plagiarisme pada kalimat berbahasa Indonesia. Pemilihan platform web diharapkan dapat memberikan kemudahan akses bagi pengguna dari berbagai perangkat, efisiensi penggunaan melalui antarmuka yang intuitif, serta

mendukung penerapan sistem deteksi plagiarisme secara luas di lingkungan pendidikan.

2.8 Penelitian Terdahulu

No	Penulis	Judul Penelitian	Hasil Penelitian
1	Salmuasih., & Sunyoto, A.	Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity	Secara umum, hasil penelitian ini menunjukkan bahwa efektivitas algoritma Rabin Karp dalam mendeteksi kemiripan dokumen dipengaruhi oleh faktor-faktor seperti banyaknya konten file, ukuran k-gram, dan proses preprocessing, di mana optimisasi terhadap aspek-aspek tersebut dapat meningkatkan performa dan akurasi sistem deteksi plagiarisme.
2	Arsad,H., Hamid, M., & Santosa, M.	Penerapan Teks Mining Dan Cosine Similarity Untuk Menentukan Kesamaan Dokumen Skripsi	Hasil penelitian menunjukkan bahwa sistem yang dibangun menggunakan metode cosine similarity mampu mengukur tingkat kemiripan judul skripsi dengan tingkat keberhasilan yang cukup akurat. Dari pengujian terhadap lima judul skripsi baru, nilai kemiripan tertinggi yang diperoleh berkisar antara 11% sampai 49%, dengan rincian sebagai berikut: judul pertama kemiripannya hanya 1%, sementara judul kedua mencapai 49%, ketiga 36%, keempat 11%, dan kelima 16%.
3	Filcha, A., & Hayaty, M.	Implementasi Algoritma Rabin-Karp	Hasil penelitian menunjukkan bahwa sistem deteksi plagiarisme berbasis

		untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa	algoritma Rabin-Karp ini berhasil mengidentifikasi tingkat kemiripan dokumen tugas mahasiswa dengan akurasi mencapai 90% berdasarkan pengujian dengan 20 pasangan dokumen. Sistem ini mampu menampilkan persentase kemiripan secara konsisten, tanpa dipengaruhi oleh urutan perbandingan dokumen. Hasil tersebut diperoleh melalui pengujian dengan confusion matrix yang menunjukkan bahwa sistem mampu mengklasifikasi tingkat kemiripan dengan baik, serta mampu membedakan dokumen yang plagiarisme ringan, sedang, dan berat berdasarkan persentase kemiripan yang sudah ditentukan
4	Ardi,S., Ahmad, Bagus, S.,Umi, Mahdiyah., Intan, N, F., & Aprisa, R, P.	Pengukuran Kemiripan Makna Menggunakan Cosine Similarity dan Basis Data Sinonim Kata	Hasil penelitian menunjukkan bahwa penggunaan ID yang didasarkan pada kelompok sinonim kata dan irisan saat proses pembobotan mampu meningkatkan nilai kemiripan makna antara dua kalimat. Dari 25 pengujian, sebanyak 24 nilai kemiripan mengalami peningkatan, dengan rata-rata nilai kemiripan mencapai 94,48%. Sebaliknya, metode atau alur pembanding memperoleh rata-rata kemiripan sekitar 69,96%. Hal ini membuktikan bahwa pendekatan berbasis basis

			data sinonim dan pengukuran vektor menggunakan cosine similarity efektif dalam menilai kemiripan makna secara lebih akurat dan konsisten.
5	Maulidya Prastita Syah., Ajeng Puspa Wardani.,M, Idhom., Trimono.	Perbandingan Representasi Teks Tf-Idf Dan Bert Terhadap Akurasi Cosine Similarity Dalam Penilaian Otomatis Jawaban Berbasis Teks	Hasil penelitian menunjukkan bahwa metode representasi teks *TF-IDF* dan *BERT* memiliki tingkat keberhasilan berbeda dalam menilai otomatis jawaban siswa. Berdasarkan metrik Cosine Similarity dan analisis statistik, *BERT* mampu menangkap makna semantik secara lebih mendalam dan akurat dibandingkan *TF-IDF*, yang lebih terbatas pada frekuensi kata dan kurang memahami konteks kalimat.

BAB III

METODOLOGI PENELITIAN

3.1. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen. Pendekatan kuantitatif dipilih karena fokus penelitian adalah pada pengukuran tingkat efisiensi dan akurasi dua metode komputasi dalam mendeteksi kesamaan teks, yaitu algoritma Rabin-Karp dan Cosine Similarity. Eksperimen dilakukan dengan cara membandingkan hasil deteksi plagiarisme pada sejumlah dokumen teks Bahasa Indonesia menggunakan kedua metode tersebut, kemudian menganalisis nilai akurasi, presisi, recall, serta waktu pemrosesan. Proses eksperimen tersebut diimplementasikan melalui sebuah sistem deteksi plagiarisme berbasis web, yang berfungsi sebagai media pengujian dan evaluasi kinerja kedua metode.

3.2. Jenis dan Sumber Data

3.2.1. Jenis Data

Data yang digunakan dalam penelitian ini berupa dokumen teks berbahasa indonesia. Dokumen tersebut mimiliki dua jenis yaitu dokumen teks bahasa indonesia asli yang belum dimodifikasi atau diubah dan dokumen teks bahasa indonesia hasil plagiasi, teks yang dibuat dengan cara menyalin secara langsung melakukan parafrasa, mengganti sinonim, atau mengubah struktur kalimat dari dokumen asli.

3.2.2. Sumber Data

Sumber data dalam penelitian ini diperoleh dari kumpulan dokumen ilmiah yang diakses melalui Semantic Scholar API. API ini digunakan untuk mengambil metadata dan konten teks artikel ilmiah secara terprogram, sehingga proses pengumpulan data dapat dilakukan secara otomatis dan terintegrasi dengan sistem deteksi plagiarisme berbasis web. Dokumen yang diambil berupa artikel ilmiah dan jurnal yang memiliki konten teks atau kalimat berbahasa Indonesia, dengan proses seleksi dilakukan berdasarkan kata kunci tertentu serta penyaringan bahasa. Dataset yang diperoleh kemudian diseleksi dan disesuaikan agar mencakup variasi panjang dokumen, yaitu dokumen pendek, sedang, dan panjang.

Selain itu, data juga diklasifikasikan berdasarkan variasi tingkat plagiarisme, meliputi copy-paste penuh, copy-paste sebagian, parafrasa, serta penggunaan sinonim. Variasi tersebut digunakan untuk menguji kinerja sistem deteksi plagiarisme dalam berbagai kondisi yang mendekati praktik plagiarisme nyata di lingkungan akademik. Data hasil pengambilan melalui API selanjutnya diproses menggunakan tahapan *preprocessing* teks Bahasa Indonesia dan dianalisis dengan algoritma Rabin-Karp dan Cosine Similarity untuk mengukur tingkat kemiripan antar dokumen.

3.3. Tahapan Penelitian

Tahapan penelitian dilakukan melalui pengembangan dan pengujian sistem deteksi plagiarisme berbasis web. Pada sistem ini, pengguna mengunggah dokumen teks atau kalimat berbahasa Indonesia melalui antarmuka web, kemudian data tersebut diproses menggunakan algoritma Rabin-Karp dan Cosine Similarity untuk mengukur tingkat kemiripan. Selanjutnya, sistem diuji untuk

memastikan kesesuaian dengan persyaratan yang telah ditetapkan, dan hasil deteksi dianalisis untuk mengetahui tingkat plagiarisme.

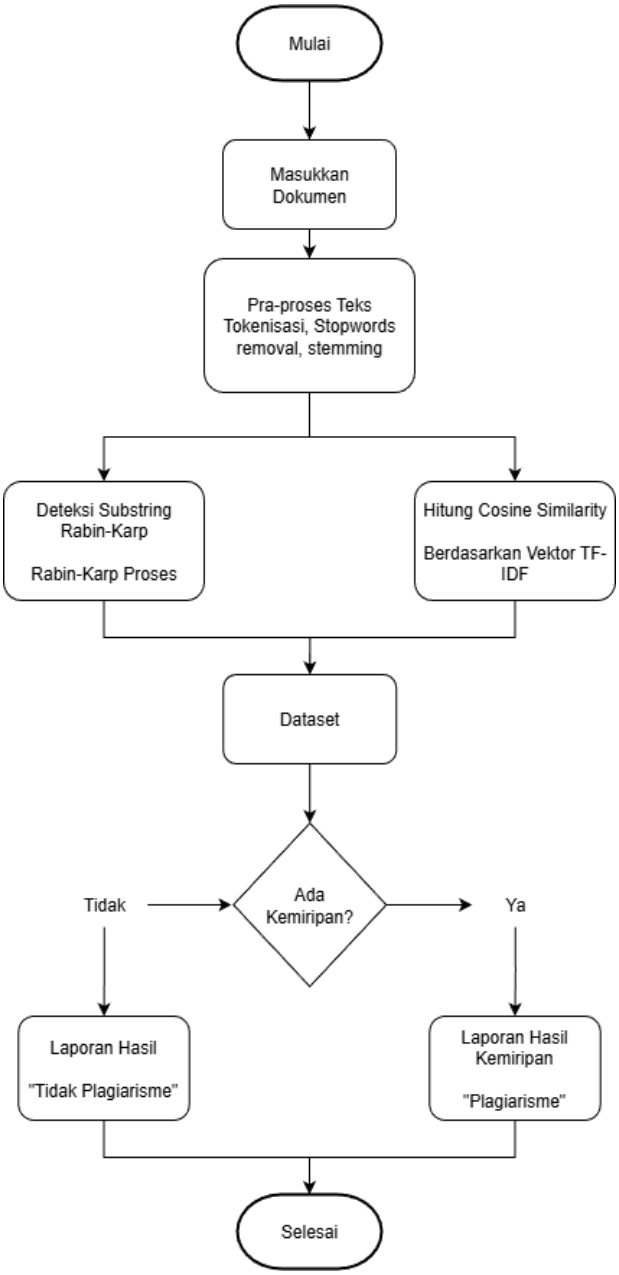
Data hasil eksperimen berupa nilai kesamaan, waktu pemrosesan, serta hasil klasifikasi plagiarisme dianalisis secara kuantitatif. Analisis dilakukan dengan menghitung rata-rata dan standar deviasi untuk menilai kinerja sistem secara keseluruhan. Hasil analisis disajikan dalam bentuk tabel atau grafik untuk memudahkan interpretasi efektivitas sistem deteksi plagiarisme yang mengombinasikan algoritma Rabin-Karp dan Cosine Similarity.

Secara keseluruhan, sistem deteksi plagiarisme ini dibangun dalam bentuk aplikasi berbasis web dengan arsitektur tiga lapisan utama, yaitu lapisan input, lapisan proses, dan lapisan output.

1. Lapisan input terdiri atas satu dokumen uji yang diunggah melalui antarmuka web serta kumpulan dokumen pembanding yang telah tersimpan didalam folder dataset sistem.
2. Lapisan Proses mencangkup tahap praproses teks atau kalimat, penerapan Rabin-Karp sebagai penyaringan awal, serta perhitungan Cosine Similarity untuk analisis kemiripan menyeluruh.
3. Lapisan Output menghasilkan nilai persentase kemiripan. Identifikasi dokumen yang paling mirip dalam dataset, serta klasifikasi tingkat plagiarisme yang kemudian dianalisis secara statistik.

Perancangan alur proses sistem deteksi plagiarisme berbasis web digambarkan dalam bentuk flowchart untuk menunjukkan tahapan kerja sistem secara sistematis, mulai dari pengguna mengunggah dokumen melalui antarmuka web

hingga sistem menghasilkan keluaran berupa tingkat kemiripan dokumen. Berikut flowchart dari sistem deteksi plagiarisme:



Gambar 3.1 Flowchart Sistem

Alur proses diawali ketika pengguna mengunggah satu dokumen kalimat yang akan diuji tingkat kemiripannya. Dokumen tersebut kemudian masuk ke tahap praproses teks untuk melakukan normalisasi data. Pada tahap ini sistem

melakukan *case folding*, *tokenisasi*, penghapusan tanda baca, *stopword removal*, dan *stemming* bahasa Indonesia agar kalimat siap dianalisis oleh algoritma.

Setelah praproses selesai, dokumen hasil olahan diproses menggunakan algoritma Rabin-Karp. Sistem membagi kalimat menjadi potongan k-gram dan menghitung nilai *hash* dengan teknik *rolling hash*. Pada tahap ini, sistem membandingkan potongan kalimat dari dokumen pengguna dengan dokumen-dokumen yang sudah tersimpan di folder dataset sebagai dokumen pembanding. Proses ini berfungsi sebagai penyaringan awal (*filtering*) untuk menemukan kemiripan *substring* secara cepat.

Selanjutnya, hasil dari proses Rabin-Karp diteruskan ke tahap perhitungan Cosine Similarity. Pada tahap ini, dokumen pengguna dan dokumen dalam dataset direpresentasikan dalam bentuk vektor berbasis TF-IDF, kemudian dihitung tingkat kemiripannya berdasarkan sudut kosinus antara vektor dokumen. Tahap ini bertujuan untuk mengukur kemiripan secara menyeluruh, termasuk pada kasus parafrase atau perubahan susunan kalimat.

Setelah nilai kemiripan diperoleh, sistem melakukan analisis terhadap tingkat plagiarisme berdasarkan persentase kemiripan yang dihasilkan. Jika nilai kemiripan tinggi, maka dokumen pengguna diindikasikan memiliki potensi plagiarisme terhadap salah satu dokumen dalam dataset. Sebaliknya, jika nilai kemiripan rendah, maka dokumen dianggap lebih orisinal.

Pada tahap akhir, sistem menampilkan hasil berupa, persentase kemiripan tertinggi dengan dokumen di dataset, dokumen mana dalam dataset yang paling mirip.Dengan demikian. flowchart perancangan alur proses ini menggambarkan secara jelas bagaimana sistem bekerja mulai dari satu dokumen input pengguna

hingga menghasilkan laporan tingkat plagiarisme berdasarkan perbandingan dengan dataset yang telah disediakan sebelumnya.

3.4. Teknik Analisis Data

Teknik analisis data pada penelitian ini dilakukan secara kuantitatif untuk menganalisis kinerja sistem deteksi plagiarisme yang mengombinasikan algoritma Rabin-Karp dan Cosine Similarity. Analisis data difokuskan pada hasil keluaran sistem berupa nilai kemiripan dokumen, waktu pemrosesan, serta hasil klasifikasi tingkat plagiarisme.

Tahapan analisis data diawali dengan pengolahan dokumen uji melalui proses pra-pemrosesan teks, yang meliputi pembersihan teks, *tokenisasi*, dan *stemming*. Tahap ini bertujuan untuk menyeragamkan bentuk teks sehingga proses pendeteksian kemiripan dapat dilakukan secara optimal.

Algoritma Rabin-Karp diterapkan sebagai tahap awal pendeteksian untuk mengidentifikasi kesamaan *substring* antar dokumen berdasarkan nilai *hash*. Hasil dari proses ini digunakan sebagai dasar penyaringan awal untuk mendeteksi kemiripan berbasis potongan teks atau kalimat.

Tahap berikutnya adalah penerapan algoritma Cosine Similarity untuk menghitung tingkat kemiripan dokumen secara keseluruhan. Pada tahap ini, dokumen direpresentasikan dalam bentuk vektor menggunakan metode TF-IDF (*Term Frequency–Inverse Document Frequency*). Nilai Cosine Similarity yang dihasilkan berada pada rentang 0 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan tingkat kemiripan yang semakin tinggi.

Hasil akhir analisis data berupa nilai kemiripan dokumen yang diperoleh dari kombinasi kedua algoritma tersebut. Data hasil eksperimen kemudian

dianalisis dengan menghitung nilai statistik deskriptif seperti rata-rata dan standar deviasi untuk menilai konsistensi dan efektivitas sistem deteksi plagiarisme. Selain itu, waktu pemrosesan dianalisis untuk mengetahui efisiensi sistem secara keseluruhan. Hasil analisis disajikan dalam bentuk tabel dan grafik untuk memudahkan interpretasi terhadap kinerja sistem dalam mendeteksi plagiarisme pada dokumen berbahasa Indonesia.

3.5.Desain Eksperimen

Desain eksperimen ini bertujuan mengukur efisiensi dan akurasi sistem deteksi plagiarisme berbasis web pada kalimat Bahasa Indonesia yang menggabungkan Rabin-Karp untuk pencocokan *substring* dan Cosine Similarity untuk kemiripan dokumen. Data eksperimen berasal dari kumpulan dokumen Bahasa Indonesia yang beragam (artikel, jurnal, tugas) dibagi menjadi himpunan referensi dan himpunan uji dengan proporsi tetap atau melalui *k-fold cross-validation* untuk kestabilan hasil, eksperimen dilakukan dengan langkah – langkah berikut :

1. Menyediakan kumpulan dataset kalimat Bahasa Indonesia, misalnya 10 dokumen asli dan 10 dokumen hasil plagiarisme dengan berbagai variasi.
2. Menjalankan sistem deteksi menggunakan algoritma Rabin-Karp pada seluruh pasangan dokumen, kemudian mencatat hasil nilai kesamaan, waktu pemrosesan, serta tingkat kesalahan.
3. Menjalankan sistem deteksi menggunakan Cosine Similarity dengan cara yang sama.
4. Membandingkan hasil kedua metode berdasarkan metrik evaluasi (akurasi, presisi, recall, dan waktu eksekusi).

3.6.Waktu dan Tempat Penelitian

3.6.1. Waktu Penelitian

Kegiatan	Bulan					
	Juni	Juli	Agustus	September	Oktober	November
Pengajuan Judul						
Observasi						
Pengumpulan Data						
Seminar Proposal						

Tabel 3. 1 Waktu Penelitian

3.6.2. Tempat Penelitian

Penelitian ini dilakukan secara *daring* (online) dengan memanfaatkan berbagai sumber terbuka di internet. Tempat penelitian tidak terbatas pada satu lokasi fisik karena proses pengumpulan data, pemrosesan, serta analisis dilakukan secara digital. Adapun rincian tempat penelitian sebagai berikut:

1. Data yang digunakan dalam penelitian ini diambil dari jurnal atau makalah tugas siswa dan mahasiswa yang menggunakan teks bahasa indonesia.
2. Waktu dan Durasi Penelitian dilakukan selama rentang waktu 14 agustus sampai 3 September.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Data

Hasil pengumpulan data pada penelitian ini berupa kumpulan dokumen kalimat berbahasa Indonesia yang digunakan sebagai dataset pembanding serta satu dokumen uji yang dianalisis menggunakan sistem deteksi plagiarisme berbasis web yang diimplementasikan dengan bahasa pemrograman Python. Dataset pembanding terdiri dari beberapa dokumen dalam format PDF dan DOCX yang telah dikumpulkan dan disimpan pada basis data sistem untuk keperluan pengujian.

Dokumen uji diperoleh dari satu dokumen teks atau kalimat yang diunggah oleh pengguna melalui antarmuka web sistem, kemudian diproses menggunakan modul pemrosesan teks berbasis Python. Dokumen tersebut dianalisis terhadap seluruh dokumen dalam dataset pembanding dengan menerapkan kombinasi algoritma Rabin-Karp dan Cosine Similarity.

Berdasarkan hasil pengumpulan data, dataset penelitian mencakup beberapa dokumen, antara lain dataset-1.pdf sampai dataset-10.pdf. Dokumen-dokumen ini berfungsi sebagai referensi pembanding dalam proses pendeteksian plagiarisme. Data yang telah terkumpul selanjutnya digunakan pada tahap pengujian sistem untuk menghasilkan nilai kemiripan dokumen serta status plagiarisme.

4.2 Implementasi Sistem Deteksi Plagiarisme

Implementasi sistem deteksi plagiarisme pada penelitian ini dilakukan dalam bentuk aplikasi berbasis web dengan bahasa pemrograman Python sebagai inti pemrosesan untuk menguji kinerja algoritma Rabin-Karp dan Cosine Similarity dalam mendeteksi tingkat kemiripan antar dokumen. Tahap ini bertujuan untuk menguji kinerja algoritma Rabin-Karp dan Cosine Similarity dalam mendeteksi tingkat kemiripan antar dokumen secara langsung melalui proses komputasi teks.

Pada tahap implementasi awal, sistem dirancang untuk membaca satu dokumen uji yang diunggah oleh pengguna melalui antarmuka web sistem dalam format PDF atau DOCX. Dokumen uji tersebut kemudian diekstraksi menjadi teks menggunakan pustaka pdfplumber untuk file PDF dan python-docx untuk file DOCX. Hasil ekstraksi teks selanjutnya diproses melalui tahap prapemrosesan yang meliputi:

a. *Case Folding*, yaitu mengubah seluruh teks atau kalimat menjadi huruf kecil guna menghindari perbedaan makna akibat variasi huruf kapital.
b. *Cleaning*, yaitu menghapus tanda baca, angka, simbol, serta karakter non-alfabet agar teks hanya berisi kata-kata relevan.
c. *Tokenisasi*, yaitu memecah teks menjadi unit kata (token) sehingga dapat dianalisis secara komputasional.
d. *Stopword Removal*, yaitu menghilangkan kata-kata umum yang tidak memiliki makna penting (seperti “dan”, “yang”, “di”, “ke”) menggunakan pustaka Sastrawi untuk bahasa Indonesia.

Setelah tahap *preprocessing*, dokumen uji diubah menjadi representasi token kata. Token-token ini kemudian dikonversi menjadi n-gram (khususnya trigram) sebagai dasar penerapan algoritma Rabin-Karp. Di sisi lain, token kata juga digunakan untuk membentuk vektor frekuensi kata dalam perhitungan Cosine Similarity.

Sistem selanjutnya melakukan proses pembandingan antara dokumen uji dengan seluruh dokumen pembanding yang tersimpan pada basis data atau direktori aplikasi web. Dataset penelitian terdiri dari beberapa dokumen referensi, yaitu dataset-1.pdf sampai dataset-10.pdf, yang telah dikumpulkan sebelumnya untuk keperluan pengujian sistem. Setiap dokumen dalam dataset diproses dengan tahapan yang sama seperti dokumen uji, mulai dari ekstraksi teks hingga *preprocessing*.

Hasil dari kedua metode kemudian digabungkan menjadi satu nilai kemiripan *hybrid* dengan bobot yang seimbang, yaitu 50% dari hasil algoritma Rabin-Karp dan 50% dari hasil Cosine Similarity. Nilai kemiripan hybrid ini digunakan sebagai dasar penentuan status plagiarisme dalam sistem. Jika skor kemiripan *hybrid* mencapai atau melebihi ambang batas 50%, maka dokumen dikategorikan sebagai “Plagiarisme”, sedangkan jika skor berada di bawah 50%, dokumen dikategorikan sebagai “Tidak Plagiarisme”.

Hasil penentuan status plagiarisme tersebut selanjutnya ditampilkan kepada pengguna melalui aplikasi berbasis web sebagai laporan hasil analisis, sehingga pengguna dapat mengetahui tingkat kemiripan dokumen secara langsung.Selama proses pengujian, sistem juga mencatat waktu komputasi untuk

menilai efisiensi kinerja algoritma. Dengan pendekatan berbasis Python ini, penelitian dapat mengevaluasi secara objektif akurasi dan kecepatan masing-masing metode.

4.3 Pengujian Sistem

Setelah pengguna menjalankan program pengujian berbasis Python, sistem akan bekerja secara otomatis tanpa campur tangan pengguna lebih lanjut. Proses otomatis yang dilakukan sistem terdiri dari beberapa tahapan utama sebagai berikut.

4.3.1. Skenario Pengujian

1. Pembacaan dokumen merupakan tahap awal dalam skenario pengujian sistem. Pada tahap ini, sistem membaca satu dokumen uji yang diunggah oleh pengguna dan disimpan di dalam folder “uji/”. Dokumen uji yang diproses dapat berupa file dengan format PDF atau DOCX. Berbeda dengan dokumen uji, dokumen pembanding tidak dimasukkan secara manual. Sistem secara otomatis mengambil data dokumen pembanding melalui *Application Programming Interface* (API) dari *https://api.semanticscholar.org/graph/v1/paper/search*. Dokumen yang diperoleh melalui API tersebut selanjutnya disimpan ke dalam folder “dataset/” dan digunakan sebagai referensi dalam proses pendeteksian plagiarisme.

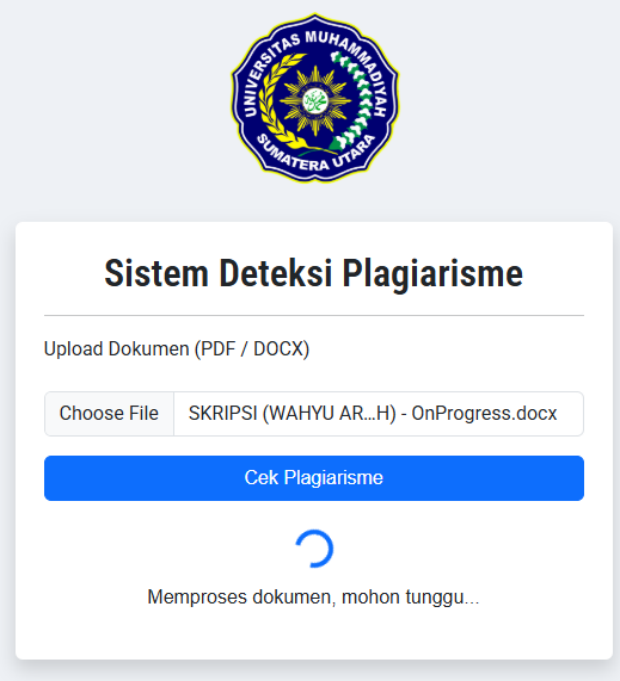

```python
# ====================
# READ FILE
# ====================
def read_file(path):
    text = ""
    try:
        if path.lower().endswith(".pdf"):
            with pdfplumber.open(path) as pdf:
                for page in pdf.pages:
                    t = page.extract_text()
                    if t:
                        text += t + " "
        elif path.lower().endswith(".docx"):
            doc = docx.Document(path)
            for p in doc.paragraphs:
                text += p.text + " "
    except:
        pass
    return text.lower()
```

```python
# ====================
# FETCH API & STORE DATASET
# ====================
def fetch_and_store_journals(query, limit=5):
    url = "https://api.semanticscholar.org/graph/v1/paper/search"
    params = {
        "query": query,
        "limit": limit,
        "fields": "title,abstract"
    }

    try:
        r = requests.get(url, params=params, timeout=10)
        data = r.json().get("data", [])

        for paper in data:
            abstract = paper.get("abstract")
            if abstract and len(abstract) > 200:
                fname = f"api_{uuid.uuid4().hex}.txt"
                with open(os.path.join(DATASET_FOLDER, fname), "w", encoding="utf8") as f:
                    f.write(abstract.lower())
    except Exception as e:
        print("API ERROR:", e)
```

Gambar 4.1 Membaca dokumen uji & Store dataset dari API

2. Pra-pemrosesan Teks (*Preprocessing*), Setelah teks berhasil diekstraksi, sistem secara otomatis melakukan pra-pemrosesan teks untuk membersihkan dan menormalisasi data. Tahapan ini melipu, *case folding* seluruh huruf diubah menjadi huruf kecil, pembersihan karakter seperti tanda baca, angka, dan simbol dihapus, tokenisasi teks atau kalimat dipecah menjadi sebuah kata-kata (token), stopword removal kata umum yang tidak bermakna seperti (“dan”, “yang”, “di”) dihapus dengan library sastrawi. Dan hasil tahap ini adalah daftar kata bersih yang siap dianalisis oleh algoritma.

```
# ====================
# PREPROCESS
# ====================
def preprocess(text):
    text = re.sub(r'[^a-zA-Z\s]', ' ', text)
    tokens = text.split()
    return [t for t in tokens if t not in STOPWORDS and len(t) > 2]
```



Gambar 4.2 Preprocessing

3. Pembentukan N-gram dan analisis Rabin-Karp), Sistem kemudian membentuk n-gram misalnya trigram atau 3-gram dari token hasil

preprocessing. Setelah itu, algoritma Rabin-Karp bekerja secara otomatis dengan Menghitung nilai hash untuk setiap n-gram dokumen uji, membandingkan nilai hash tersebut dengan n-gram pada setiap dokumen didataset, menghitung persentase kemiripan berdasarkan jumlah substring yang cocok, tahap ini berfungsi sebagai penyaringan awal (*filtering*) untuk mendeteksi kemiripan yang *substring* secara cepat.

```
# ====================
# NGRAM
# ====================
def ngrams(tokens, n=3):
    return [" ".join(tokens[i:i+n]) for i in range(len(tokens)-n+1)]

# ====================
# RABIN-KARP
# ====================
def rabin_karp(ng1, ng2):
    s1, s2 = set(ng1), set(ng2)
    if not s1 or not s2:
        return 0
    return (len(s1 & s2) / len(s1)) * 100
```

```
N-Gram Uji: ''contoh kalimat pengujian', 'kalimat pengujian mendeteksi, "pengujian
         , mendeteksi plagiarisme"

Rabin-Karp: dataset-2.docx - Match: 60.0%
```

Gambar 4.3 N-gram dan Rabin-Karp

4. Perhitungan Cosine Similarity, setalah tahap Rabin-Karp selesai sistem otomatis melanjutkan ke perhitungan Cosine Similarity dengan langkah mengubah dokumen uji dan dokumen dataset menjadi representasi vektor menggunakan frekuensi kata, menghitung sudut kosinus antara kedua vektor dokumen, menghasilkan nilai kemiripan antara 0-100%. Tahap ini bertujuan untuk menangkap kemiripan makna teks atau kalimat, termasuk jika terjadii parafrase atau perubahan susunan kalimat.

```python
# ====================
# COSINE SIMILARITY
# ====================
def cosine(t1, t2):
    f1, f2 = Counter(t1), Counter(t2)
    common = set(f1) & set(f2)
    dot = sum(f1[w] * f2[w] for w in common)
    mag1 = math.sqrt(sum(v*v for v in f1.values()))
    mag2 = math.sqrt(sum(v*v for v in f2.values()))
    return 0 if mag1 == 0 or mag2 == 0 else (dot / (mag1 * mag2)) * 100
```

```
Menghitung Cosine Similarity...

Cosine Similarity: dataset-2.docx - Score: 55.23%
```

Gambar 4.4 Cosine Similarity

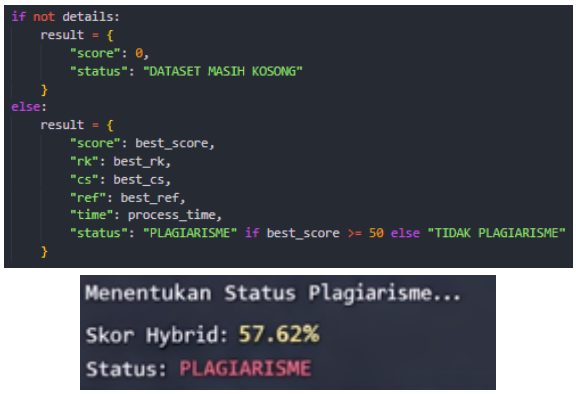
5. Penggabungan Skor (*Hybrid Score*), sistem kemudian mengkombinasikan hasil kedua metode menggunakan rumus sederhana. Skor hybrid ini merepresentasikan tingkat kemiripan akhir antara dokumen uji dan setiap dokumen dalam dataset.

```python
for f in os.listdir(DATASET_FOLDER):
    if f.lower().endswith(ALLOWED_EXT):
        ref_path = os.path.join(DATASET_FOLDER, f)
        text_ref = read_file(ref_path)
        tokens_ref = preprocess(text_ref)

        if len(tokens_ref) < 30:
            continue

        rk = round(rabin_karp(ngrams(tokens_uji), ngrams(tokens_ref)), 2)
        cs = round(cosine(tokens_uji, tokens_ref), 2)
        hybrid = round((0.5 * rk) + (0.5 * cs), 2)

        details.append({
            "ref": f,
            "rk": rk,
            "cs": cs,
            "hybrid": hybrid
        })

        if hybrid > best_score:
            best_score = hybrid
            best_rk = rk
            best_cs = cs
            best_ref = f

process_time = round(time.time() - start_time, 2)

os.remove(upload_path)
```

```
Menghitung Skor Hybrid...
Rabin-Karp: 60.0%
Cosine Similarity: 55.23%
Skor Hybrid: 57.62%
```

Gambar 4.5 Penggabungan Skor (Hybrid Score)

6. Penentuan Status Plagiarisme, berdasarkan skor hybrid sistem secara otomatis menentukan status dokumen uji dengan kriteria, jika skor ≥ 50% = Plagiarisme, jika skor < 50% = Tidak Plagiarisme.



Gambar 4.6 Penentuan Status Plagiarisme

7. Output Hasil Pengujian, pada tahap akhir, sistem menampilkan hasil secara otomatis, seperti nama dokumen uji, nama dokumen dataset paling mirip, nilai Rabin-Karp(%), nilai Cosine Similarity (%), skor hybrid (%) waktu pemrosesan (detik/ms), status plagiarisme.

Gambar 4.7 Output Hasil Pengujian

Dengan mekanisme ini, sistem mampu melakukan seluruh proses deteksi plagiarisme secara otomatis mulai dari pembacaan dokumen, preprocessing teks atau kalimat, analisis Rabin-Karp dan Cosine Similarity. Hingga penetuan status plagiarisme tanpa intervensi manual dari pengguna.

4.3.2. Parameter Pengujian

No	Parameter	Nilai/Spesifikasi	Keterangan
1	Bahasa dokumen	Bahasa Indonesia	Fokus penelitian pada kalimat berbahasa indonesia
2	Format dokumen	PDF dan DOCX	Format yang diproses sistem
3	Jumlah dokumen uji	satu dokumen per pengujian	Satu dokumen dibandingkan ke dataset
4	Jumlah dataset pembanding	10 dokumen	Dataset-1.pdf sampai dataset-10.pdf
5	Teknik preprocessing	Case folding, cleanning, tokenisasi, stopword removal	Menggunakan pustaka sastrawi
6	Ukuran n-gram	Trigram (3-gram)	Untuk algoritma Rabin-Karp
7	Metode kemiripan	Rabin-Karp dan Cosine Similarity	Kombinasi dua metode
8	Bobot hybrid	50% Rabin-Karp:50% Cosine Similarity	Bobot seimbang
9	Ambang batas plagiarisme	≥ 50%	Diatas nilai ini dianggap plagiarisme

10	Output utama	Skor Rabin-Karp, Cosine Similarity, hybrid, waktu	Ditampilkan pada output terminal python

Tabel 4.1 Parameter Pengujian

Pengujian dilakukan berbasis Python dengan satu dokumen uji yang dibandingkan terhadap 10 dokumen dalam dataset pembanding. Sistem menerapkan preprocessing teks, membentuk trigram untuk Rabin-Karp, serta menghitung Cosine Similarity berbasis vektor kata. Kedua metode digabungkan dengan bobot seimbang (50:50) dan ambang batas plagiarisme ditetapkan sebesar 50%.

4.4 Pembahasan

Berdasarkan hasil implementasi dan pengujian sistem deteksi plagiarisme berbasis Python, penelitian ini menunjukkan bahwa kombinasi algoritma Rabin-Karp dan Cosine Similarity mampu memberikan pendekatan yang lebih komprehensif dalam mendeteksi kemiripan kalimat berbahasa Indonesia dibandingkan penggunaan satu metode secara terpisah.

Algoritma Rabin-Karp berperan efektif sebagai tahap penyaringan awal (*filtering*) dengan memanfaatkan teknik *rolling hash* dan n-gram untuk mendeteksi kemiripan *substring* secara cepat. Metode ini terbukti efisien dalam mengidentifikasi kesamaan teks atau kalimat yang bersifat eksak atau hampir identik, terutama pada kasus plagiarisme berbasis copy-paste. Namun, hasil pengujian juga menunjukkan bahwa Rabin-Karp memiliki keterbatasan dalam menangkap kemiripan semantik, khususnya pada kalimat yang mengalami parafrase atau perubahan struktur kalimat.

Untuk mengatasi keterbatasan tersebut, penerapan Cosine Similarity berbasis representasi vektor TF-IDF memberikan kontribusi penting dalam mengukur

kemiripan makna antar dokumen. Metode ini lebih sensitif terhadap kesamaan isi meskipun terjadi perubahan susunan kata atau penggunaan sinonim. Dengan demikian, Cosine Similarity melengkapi kelemahan Rabin-Karp dan meningkatkan kemampuan sistem dalam mendeteksi plagiarisme yang tidak bersifat literal.

Penggabungan kedua metode dalam bentuk skor hybrid terbukti menghasilkan evaluasi kemiripan yang lebih stabil dan representatif. Skor hybrid tidak hanya mempertimbangkan kecocokan *substring*, tetapi juga kesamaan makna secara keseluruhan. Hal ini membuat sistem lebih adaptif dalam menangani berbagai bentuk plagiarisme, baik yang bersifat langsung maupun parafrase.

Dari sisi kinerja, hasil pengujian menunjukkan bahwa waktu pemrosesan sistem masih berada dalam kategori efisien, meskipun meningkat seiring bertambahnya jumlah dokumen dalam dataset. Hal ini disebabkan oleh proses pembandingan dokumen uji dengan seluruh dataset serta perhitungan TF-IDF pada tahap Cosine Similarity. Oleh karena itu, sistem ini lebih cocok digunakan pada skala menengah, dan untuk skala besar diperlukan optimasi lebih lanjut, misalnya dengan indexing dokumen atau pemrosesan paralel.

Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi Rabin-Karp dan Cosine Similarity merupakan pendekatan yang layak diterapkan dalam sistem deteksi plagiarisme kalimat berbahasa Indonesia. Sistem yang dikembangkan mampu memberikan hasil yang informatif, transparan, serta relevan untuk mendukung integritas akademik dan mencegah praktik plagiarisme dalam penulisan ilmiah.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem deteksi plagiarisme teks berbahasa Indonesia menggunakan kombinasi algoritma Rabin-Karp dan Cosine Similarity, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem berhasil dirancang dan diimplementasikan berbasis Python untuk mendeteksi tingkat kemiripan teks atau kalimat antara satu dokumen uji dengan dokumen-dokumen dalam dataset pembanding. Sistem mampu membaca dokumen berformat PDF dan DOCX, melakukan *preprocessing* teks, serta menghitung kemiripan secara otomatis tanpa intervensi manual pengguna.
2. Tahap *preprocessing* teks berperan penting dalam meningkatkan kualitas hasil deteksi. Proses *case folding*, *cleaning*, *tokenisasi,* dan

stopword removal menggunakan pustaka Sastrawi terbukti membantu sistem bekerja lebih stabil dan mengurangi noise pada teks atua kalimat sebelum dianalisis oleh algoritma.

3. Algoritma Rabin-Karp efektif sebagai tahap penyaringan awal (*filtering*) untuk mendeteksi kemiripan berbasis *substring* (n-gram), terutama pada kasus plagiarisme yang bersifat literal atau *copy-paste*. Namun, metode ini memiliki keterbatasan dalam menangkap kemiripan makna ketika terjadi parafrase.
4. Cosine Similarity mampu melengkapi kelemahan Rabin-Karp dengan mengukur kemiripan semantik berbasis representasi vektor kata (TF-IDF). Metode ini lebih sensitif terhadap perubahan susunan kalimat dan penggunaan sinonim.
5. Penggabungan kedua metode dalam bentuk skor hybrid (50:50) menghasilkan ukuran kemiripan yang lebih komprehensif dan stabil dibandingkan penggunaan satu metode secara terpisah. Skor hybrid mampu menangkap kemiripan baik secara sintaksis maupun semantik.
6. Sistem dapat mengklasifikasikan dokumen secara otomatis ke dalam kategori Plagiarisme atau Tidak Plagiarisme berdasarkan ambang batas 50%, serta menampilkan dokumen dataset yang paling mirip beserta waktu pemrosesan.
7. Dari sisi kinerja, sistem menunjukkan waktu komputasi yang masih efisien untuk skala dataset menengah, namun berpotensi meningkat seiring bertambahnya jumlah dokumen pembanding.

Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi Rabin-Karp dan Cosine Similarity merupakan pendekatan yang layak dan efektif untuk mendeteksi plagiarisme pada kalimat atau teks berbahasa Indonesia.

5.2 Saran

Berdasarkan keterbatasan dan hasil penelitian yang telah dilakukan, beberapa saran untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Optimasi performa untuk dataset besar untuk skala dokumen yang lebih banyak, disarankan menggunakan teknik indexing dokumen atau pemrosesan paralel agar waktu komputasi lebih cepat.
2. Eksperimen dengan ukuran N-gram berbeda penelitian lanjutan dapat menguji berbagai ukuran n-gram (bi-gram, tri-gram, atau 4-gram) untuk melihat pengaruhnya terhadap akurasi Rabin-Karp.
3. Penggunaan metode semantik yang lebih lanjut disarankan untuk mengombinasikan sistem ini dengan metode berbasis *word embedding* seperti *Word2Vec*, *FastText*, atau *BERT* agar mampu mendeteksi parafrase lebih kompleks.
4. Penetapan ambang batas yang lebih dinamis ambang batas 50% dapat dievaluasi kembali dengan pengujian pada lebih banyak data untuk memperoleh nilai *threshold* yang lebih optimal.
5. Pengujian dengan dataset berlabel untuk meningkatkan validitas penelitian, disarankan menggunakan dataset yang sudah diberi label plagiarisme dan non-plagiarisme sehingga akurasi sistem dapat dievaluasi lebih objektif.

DAFTAR PUSTAKA

Salmuasih, & Sunyoto, A. (2013). Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity. *Jurnal Portal – Universitas Islam Indonesia*

Turnitin. (2023). Turnitin AI Detection Feature reviews more than 65 million papers.

Neliti. (2018) Pendeteksian plagiarisme menggunakan algoritma rabin-karp dengan metode rolling. *Jurnal Informatika*, 3(1), 39-45

Filcha, A., & Hayaty, M. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. JUITA: *Jurnal Informatika*, 7(1), 25–32.

Mulyana, (2010). Pencegahan Tindak Plagiarisme Dalam Penulisan Skripsi Upaya Memperkuat Pembentukan Karakter Di Dunia Akademik. *Jurnal Cakrawala Pendidikan*, 1(3).

Purwitasari, D., Kusmawan, P.Y. & Yuhana, U.L., 2010. Deteksi Keberadaan Kalimat Sama sebagai Indikasi Penjiplakan dengan Algoritma Hashing Berbasis N-Gram. Surabaya: *Institut Teknologi Sepuluh Nopember*.

Gusnayetti, G. (2025). *Dampak Plagiarisme Terhadap Penulisan Artikel Ilmiah. Jurnal Penelitian Dan Pengkajian Ilmiah Eksakta, 4(1), 122-130.*

Arsad, H., Hamid, M., & Santosa, M. (2024). Penerapan Teks Mining Dan Cosine Similarity Untuk Menentukan Kesamaan Dokumen Skripsi. *Indonesian Journal On Information System*, 9(1), 1-9.

Maulidya Prastita Syah., Ajeng Puspa Wardani., M, Idhom., Trimono. (2025). Perbandingan Representasi Teks Tf-Idf Dan Bert Terhadap Akurasi Cosine Similarity Dalam Penilaian Otomatis Jawaban Berbasis Teks. *Jurnal Data Sciences Indonesia*, 5(1), 47–59.

Setiawan, A., Indah Fitri A., Awang Harsa K. (2015). Klasifikasi Dan Pencarian Buku Referensi Akademik Menggunakan Metode Naïve Bayes Classifier (NBC) (Studi Kasus: Perpustakaan Daerah Provinsi Kalimantan Timur). *Jurnal Informatika Mulawarman*, 10(1), 1-10.

Yuniar, E., Dwi Safiroh., Dian Wahyuningsih. (2022). Implementasi Scraping Data Untuk Sentiment Analysis Pengguna Dompet Digital Dengan Menggunakan Algoritma Machine Learning. *Jurnal Janitra Informatika Dan Sistem Informasi*, 2(1), 35-42.

Alun, S, I., Anggun, F. (2021). Implementasi Metode Vector Space Model Untuk Deteksi Emosi Menggunakan Data Teks Twitter. *Jurnal Restikom: Riset Teknik Informatika dan Komputer*, 3(3), 116-129.

Raja Farhan, R., Dian Eka, R., & Issa A. (2022). Implementasi Algoritma Support Vector Machine dan Model *Bag-of-Words* dalam Analisis Sentimen mengenai PILKADA 2020 pada Pengguna Twitter. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 6(10), 4924–4931.

Ardi, S., Ahmad, Bagus, S., Umi, Mahdiyah., Intan, N, F., & Aprisa, R, P., (2023). Pengukuran Kemiripan Makna Menggunakan Cosine Similarity dan Basis Data Sinonim Kata. *Jurnal Teknologi dan Ilmu Komputer,* 10(4), 747–752.

Ari, Kurniawan, S., Robby, Yuli, E., Fenty, Arani., Tia, Tanjung., & Agustian, Prakarsya., (2023). Implementasi Algoritma Rabin-Karp pada Pendeteksian Plagiarisme. *Jurnal Management Sistem Informasi dan Teknologi,* 13(1), 23–24.

Herianto, Yulisman, Winda., Herianti., & M, Yuda, Irawan., (2021). Aplikasi Deteksi Plagiarisme Judul Tugas Akhir Berbasis Web Dengan Menggunakan Algoritma Rabin-Karp Rolling Hash (Studi Kasus:AMIK MAHAPUTRA RIAU). *Jurnal Ilmu Komputer,* 10(2), 107–112.

Hardison, & Maulana., Ardhiansyah, (2023). Deteksi Plagiarisme Pada File Dokumen Berdasarkan Tingkat Kesamaan Dengan Menggunakan Metode Algoritma Rabin-Karp Berbasis Web. *Jurnal Ilmu Komputer dan Science*, 2(3), 760–766.

LAMPIRAN

Lampiran 1

I. Tujuan Perbandingan

Lampiran ini bertujuan untuk menyajikan data pendukung secara rinci terkait hasil perbandingan antara sistem deteksi plagiarisme yang dikembangkan dengan perangkat lunak Turnitin. Perbandingan dilakukan untuk menunjukkan keunggulan sistem yang dikembangkan pada aspek-aspek yang dapat diukur dan dianalisis secara akademik, seperti transparansi algoritma, kontrol terhadap dataset pembanding, efisiensi waktu pemrosesan, serta fleksibilitas dalam pengujian parameter. Turnitin digunakan sebagai pembanding secara konseptual karena merupakan perangkat lunak komersial dengan algoritma dan basis data yang bersifat *proprietary* (hak milik), sehingga peneliti tidak memiliki akses langsung untuk melakukan pengujian dan analisis internal sistem tersebut.

Oleh karena itu, perbandingan dalam penelitian ini tidak difokuskan pada kesamaan algoritma atau replikasi hasil Turnitin, melainkan pada analisis karakteristik dan performa sistem yang dikembangkan berdasarkan data hasil pengujian. Pendekatan ini diharapkan dapat memberikan gambaran objektif mengenai kelebihan sistem dalam konteks penelitian akademik serta mendukung validitas hasil yang disajikan.

II. Ringkasan Perbandingan

Aspek Perbandingan	Sistem Deteks Plagiarisme website	Turnitin
Transparansi algoritma	Tinggi	Rendah
Kontrol dataset	Dapat dikendalikan	Tidak dapat dikendalikan

Waktu pemrosesan	Terukur	Tidak tersedia
Fleksibilatas pengujian	Tinggi	Rendah
Tujuan penggunaan	Penelitian akademik	Komersial

Tabel Ringkasan Perbandingan

Berdasarkan tabel ringkasan perbandingan antara sistem deteksi plagiarisme berbasis web dan Turnitin, dapat disimpulkan bahwa masing-masing sistem memiliki karakteristik dan tujuan penggunaan yang berbeda. Turnitin merupakan perangkat lunak deteksi plagiarisme komersial yang dirancang untuk penggunaan praktis dengan cakupan basis data yang luas, namun memiliki keterbatasan dari sisi transparansi dan fleksibilitas pengujian. Sebaliknya, sistem yang dikembangkan dalam penelitian ini lebih berorientasi pada kebutuhan penelitian akademik, sehingga menekankan aspek keterukuran, keterbukaan, dan kemudahan analisis.

Dari aspek transparansi algoritma, sistem deteksi plagiarisme berbasis web memiliki keunggulan yang signifikan karena menggunakan metode Rabin-Karp dan Cosine Similarity yang bersifat terbuka dan dapat dijelaskan secara matematis. Seluruh tahapan proses pendeteksian plagiarisme dapat dianalisis, diuji ulang secara ilmiah. Hal ini berbeda dengan Turnitin yang menggunakan algoritma proprietary, sehingga proses internal perhitungan kemiripan tidak dapat diketahui atau dianalisis lebih lanjut oleh pengguna. Tingginya transparansi algoritma pada sistem deteksi plagiarisme berbasis web menjadi nilai tambah dalam konteks penelitian dan pengembangan metode.

Pada aspek kontrol dataset, sistem deteksi plagiarisme berbasis web juga menunjukkan keunggulan dibandingkan Turnitin. Sistem yang dikembangkan

memungkinkan peneliti untuk menentukan secara langsung sumber, jenis, dan jumlah dokumen pembanding melalui mekanisme pengambilan data menggunakan API. Dengan adanya kontrol dataset ini, pengujian dapat dilakukan secara terarah dan sesuai dengan kebutuhan penelitian. Sementara itu, Turnitin menggunakan basis data global yang bersifat tertutup, sehingga pengguna tidak memiliki kendali terhadap dokumen pembanding yang digunakan dalam proses pendeteksian. Kondisi ini membatasi Turnitin untuk keperluan eksperimen yang membutuhkan dataset terdefinisi dengan jelas.

Dari aspek waktu pemrosesan, sistem deteksi plagiarisme berbasis web memiliki kelebihan karena waktu proses pendeteksian dapat diukur secara langsung dan disajikan dalam bentuk data kuantitatif. Waktu pemrosesan dapat dianalisis berdasarkan variasi jumlah dataset, sehingga kinerja sistem dapat dievaluasi secara objektif. Turnitin tidak menyediakan informasi waktu pemrosesan secara rinci kepada pengguna, sehingga aspek efisiensi sistem tidak dapat dianalisis lebih lanjut. Oleh karena itu, keterukuran waktu pemrosesan menjadi salah satu keunggulan sistem usulan dalam evaluasi performa.

Dari sisi fleksibilitas pengujian, sistem yang dikembangkan memiliki tingkat fleksibilitas yang tinggi karena memungkinkan variasi parameter, seperti nilai k-gram dan ambang batas kemiripan. Fleksibilitas ini memungkinkan peneliti untuk melakukan pengujian eksperimental dan menganalisis pengaruh perubahan parameter terhadap hasil deteksi plagiarisme. Sebaliknya, Turnitin tidak menyediakan akses untuk mengubah parameter internal sistem, sehingga fleksibilitas pengujian relatif rendah dan kurang mendukung kebutuhan penelitian eksperimental.

Sistem deteksi plagiarisme berbasis web dan Turnitin memiliki orientasi yang berbeda. Sistem usulan dikembangkan khusus untuk mendukung kegiatan penelitian akademik, pembelajaran, dan pengembangan metode deteksi plagiarisme. Sementara itu, Turnitin dirancang sebagai perangkat lunak komersial yang berfokus pada layanan pendeteksian plagiarisme untuk institusi pendidikan dan publikasi. Perbedaan tujuan ini menjelaskan mengapa sistem deteksi plagiarisme berbasis web lebih unggul pada aspek transparansi, fleksibilitas, dan keterukuran, sedangkan Turnitin unggul pada sisi penggunaan praktis dan cakupan basis data.

Secara keseluruhan, dapat disimpulkan bahwa meskipun Turnitin memiliki keunggulan sebagai sistem deteksi plagiarisme komersial dengan basis data yang luas, sistem yang dikembangkan dalam penelitian ini lebih sesuai untuk kebutuhan penelitian akademik karena menyediakan transparansi algoritma, kontrol dataset, waktu pemrosesan yang terukur, serta fleksibilitas pengujian yang tinggi. Dengan demikian, sistem deteksi plagiarisme berbasis web ini dapat dijadikan alternatif yang relevan dan valid dalam konteks penelitian dan pengembangan sistem deteksi plagiarisme.

Lampiran 2

Surat Penetapan Dosen Pembimbing

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU
Unggul | Cerdas | Terpercaya

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 695/II.3-AU/UMSU-09/F/2025

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : **Teknologi Informasi**
Pada tanggal : **25 Juni 2025**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : **Wahyu Ardiansyah**
NPM : **2109020120**
Semester : **VIII (Delapan)**
Program studi : **Teknologi Informasi**
Judul Proposal / Skripsi : **DETEKSI PLAGIARISME TEKS BAHASA INDONESIA MENGGUNAKAN ALGORITMA RABIN-KARP DAN COSINE SIMILARITY EFISIENSI DAN AKURASI**

Dosen Pembimbing : **Dr. Zainal Azis, M.Si**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **25 Juni 2026**
4. Revisi judul……

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 29 Dzulhijjah 1446 H
25 Juni 2025M

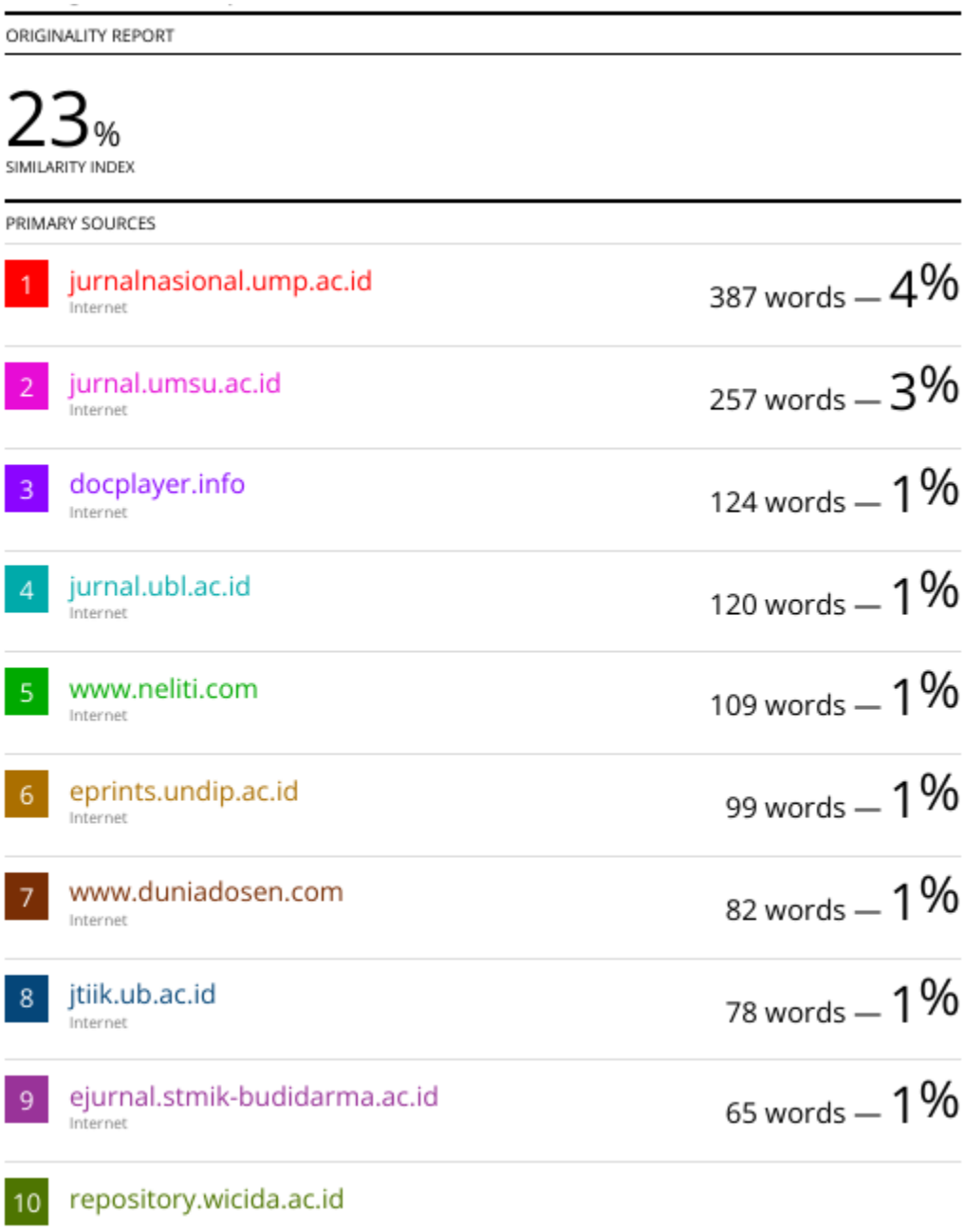
Dekan

Dr. Al-Khowarizmi, M.Kom.
NIDN : 0127099201

Cc. File

BAN-PT MQA QS STARS

Cek Turnitin



ORIGINALITY REPORT

23%
SIMILARITY INDEX

PRIMARY SOURCES

1	jurnalnasional.ump.ac.id Internet	387 words — 4%
2	jurnal.umsu.ac.id Internet	257 words — 3%
3	docplayer.info Internet	124 words — 1%
4	jurnal.ubl.ac.id Internet	120 words — 1%
5	www.neliti.com Internet	109 words — 1%
6	eprints.undip.ac.id Internet	99 words — 1%
7	www.duniadosen.com Internet	82 words — 1%
8	jtiik.ub.ac.id Internet	78 words — 1%
9	ejurnal.stmik-budidarma.ac.id Internet	65 words — 1%
10	repository.wicida.ac.id	