PROPOSAL SKRIPSI

DETEKSI PLAGIARISME TEKS BAHASA INDONESIA MENGGUNAKAN ALGORITMA RABIN-KARP DAN COSINE SIMILARITY EFISIENSI DAN AKURASI

DISUSUN OLEH:

WAHYU ARDIANSYAH

NPM.2109020120

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2025

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat telah mempermudah masyarakat dalam mengakses, mengolah, dan mendistribusikan informasi, khususnya dalam bentuk teks. Di satu sisi, kemudahan ini memberikan manfaat besar dalam dunia pendidikan, penelitian, dan publikasi. Namun, di sisi lain, hal ini juga memicu peningkatan praktik plagiarisme, yaitu tindakan menjiplak karya orang lain tanpa memberikan atribusi yang semestinya.

Plagiarisme menjadi masalah serius dalam dunia pendidikan dan akademik, terutama dengan kemudahan akses informasi dan copy-paste melalui internet. Deteksi plagiarisme tidak hanya berguna untuk mengungkap ketidakaslian, tetapi juga mendukung integritas akademik dan kejujuran ilmiah.

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 teks berbahasa Indonesia adalah kompleksitas bahasa itu sendiri, yang memiliki struktur morfologi kaya dan variasi sinonim yang luas. Deteksi plagiarisme membutuhkan metode yang mampu mengidentifikasi kesamaan baik secara eksak (tepat sama) maupun parafrase (makna sama dengan kata berbeda).

Algoritma **Rabin–Karp** menggunakan *rolling hash* untuk menyapu teks panjang dan membandingkan potongan (k-gram) secara cepat. Secara praktik, pendekatan ini efisien untuk penyaringan awal (filtering) karena biaya hashing linear terhadap panjang teks, sedangkan verifikasi karakter-demi-karakter hanya terjadi saat hash cocok. implementasi **Rabin–Karp** pada sistem cek plagiarisme menunjukkan akurasi 90% pada 20 percobaan dibandingkan Plagiarism Checker X (komparasi nilai kemiripan pada pasangan dokumen). Hasil ini menegaskan kegunaan **Rabin–Karp** sebagai modul deteksi cepat untuk kesamaan berbasis n-gram.

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).

Dari uraian di atas, penulis mengangkat topik **Deteksi Plagiarisme Teks Bahasa Indonesia Menggunakan Algoritma Rabin-Karp dan Cosine Similarity** sebagai upaya mengkaji penerapan teknologi pemrosesan teks (text processing) dan pemrosesan bahasa alami. Penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan ilmu pengetahuan di bidang

pemrosesan teks, serta memberikan solusi praktis bagi para karya ilmiah dalam mengurangi plagiarisme dalam teks Bahasa 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 teks?
2. Bagaimana performa sistem dalam hal efisiensi waktu dan akurasi deteksi?
3. Apakah kombinasi kedua algoritma tersebut dapat meningkatkan kualitas deteksi plagiarisme?

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 teks berbahasa Indonesia.
2. Dataset berupa dokumen tugas, artikel, atau skripsi pendek.
3. Pemrosesan terbatas pada teks tertulis (tidak mencakup gambar atau tabel).
4. Implementasi menggunakan Python dan pustaka pendukung seperti scikit-learn, nltk, atau Sastrawi.

1.4 Tujuan Penelitian

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

1. Mengimplementasikan algoritma **Rabin-Karp** untuk deteksi substring kemiripan dalam teks.
2. Mengimplementasikan **Cosine Similarity** berbasis representasi vector dokumen.
3. Menganalisis dan membandingkan efisiensi serta akurasi dari masing – masing metode.
4. Menggabungkan keduanya untuk menciptakan sistem deteksi plagiarisme yang lebih optimal.

1.5 Manfaat Penelitian

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

1. Memberikan solusi teknologi 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 *stop word* 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 (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 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	Hasil penelitian menunjukkan bahwa sistem yang dibangun menggunakan metode cosine similarity mampu

		Kesamaan Dokumen Skripsi	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 untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa	Hasil penelitian menunjukkan bahwa sistem deteksi plagiarisme berbasis 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.

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 diperoleh dari beberapa artikel jurnal, skripsi, berita daring, serta teks buatan peneliti sendiri untuk kepentingan eksperimen. Data dipilih agar memiliki variasi panjang dokumen (pendek, sedang, panjang) serta variasi tingkat plagiarisme (copy-paste penuh, copy sebagian, parafrasa, dan penggunaan sinonim). Dengan demikian, sistem dapat diuji pada berbagai kondisi nyata yang sering terjadi dalam praktik plagiarisme.

3.3. Tahapan Penelitian

Tahap penelitian dilakukan dengan mengumpulkan berbagai dokumen atau tesis, kemudian data tersebut diolah dengan menerapkan algoritma Rabin-Karp dan Cosine Similarity untuk mengukur tingkat kemiripan, setelah itu sistem pengujian dilakukan untuk memastikan kesesuain dengan persyaratan, dan hasil deteksi dianalisis untuk mengetahui tingkat plagiarisme. Untuk metode penelitian ini dibagi dalam beberapa tahapan utama sebagai berikut :

a. Pra-pemrosesan Teks (Text Preprocessing)

Sebelum dilakukan perhitungan kesamaan, dokumen teks harus melalui proses pra-pemrosesan untuk memastikan bahwa data sudah bersih. Tahap ini juga mencakup tokenisasi untuk memecah teks menjadi potongan kata, lalu stemming mengembalikan kata ke bentuk dasarnya, misalnya “menyanyi” menjadi “nyanyi”, “berlari” menjadi “lari”.

b. Penerapan Algoritma Rabin-Karp

Algoritma Rabin-Karp digunakan untuk mendeteksi kesamaan pola antar dokumen dengan cara menghitung *hash value* dari *substring text*, dengan proses membagi teks dokumen menjadi potongan – potongan *substring* dengan panjang tertentu, lalu menghitung nilai *hash* dari setiap *substring* dan mencocokan nilai *hash* antar dokumen untuk menemukan *substring* yang identik.

c. Penerapan Cosine Similarity

Algoritma Cosine Similarity digunakan untuk perhitungan kesamaan suatu teks dokumen, mengubah dokumen menjadi *representasi* vektor menggunakan metode *TF-IDF (Term Frequency- Inverse Document Frequency)*, kemudian menghitung nilai Cosine Similarity, dan hasil perhitungan berupa nilai 0 sampai 1 semakin tinggi tingkat kesamaanya antar dokumen.

d. Pengujian Efisiensi dan Akurasi

Untuk mengukur kinerja kedua metode, dilakukan pengujian dengan beberapa metrik evaluasi, seperti efisiensi (waktu komputasi) diukur berdasarkan waktu eksekusi algoritma dalam membandingkan dua dokumen. Dan akurasi diukur berdasarkan kesesuaian hasil deteksi

plagiarisme dengan kondisi sebenarya (*ground truth*). Kemudian presisi dan recal digunakan untuk menilai sejauh mana metode mampu mendeteksi plagiarisme dengan benar dan tanpa banyak kesalahaan. Hasil pengujian kemudian dibandingkan sehingga dapat diketahui apakah Rabin-Karp lebih efisien dibandingkan Cosine Similarity lebih akurat dibandingkan Rabin-Karp, atau sebaliknya.

3.4. Desain Eksperimen

Desain eksperimen ini bertujuan mengukur efisiensi dan akurasi sistem deteksi plagiarisme teks 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, esai, 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 teks Bahasa Indonesia, misalnya 20 dokumen asli dan 20 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).
5. Melakukan analisis perbandingan untuk menyimpulkan kelebihan dan kekurangan kedua metode.

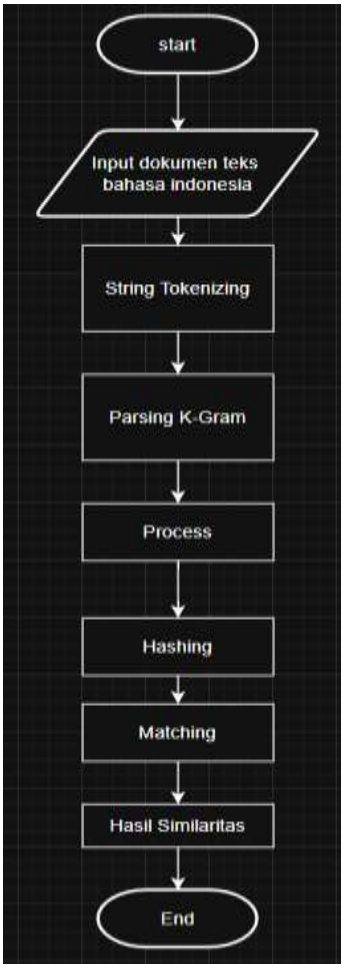
3.5. Teknik Analisis Data

Data hasil eksperimen berupa nilai kesamaan, waktu pemrosesan, serta hasil klasifikasi plagiarisme dianalisis secara kuantitatif. Analisis dilakukan dengan menghitung rata-rata, standar deviasi, serta perbandingan hasil antara dua metode.

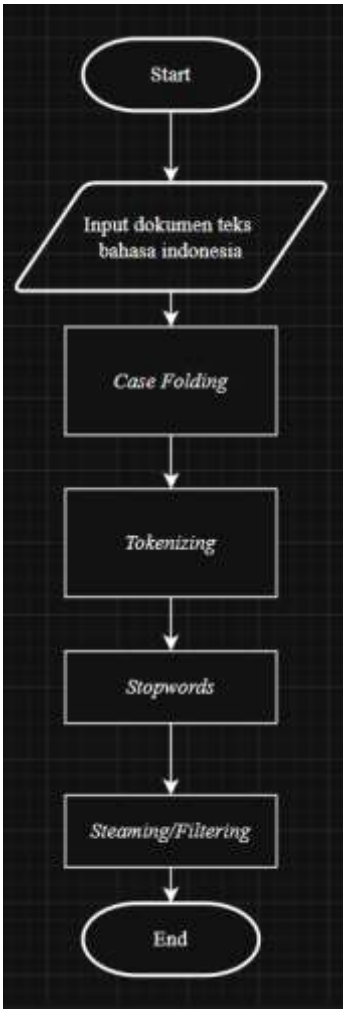
Selain itu, dilakukan penggambaran grafik atau tabel untuk memudahkan interpretasi mengenai perbedaan efisiensi dan akurasi.

Teks analisis data pada penelitian ini menggunakan algoritma Rabin-Karp dan Cosine Similarity dan difokuskan pada evaluasi akurasi hasil deteksi serta efisiensi proses pencarian kemiripan dokumen. Setelah data teks melalui tahap praproses berupa tokenisasi, *parsing K-Gram*, *process*, *hashing*, *matching*, dan hasil similaritas.

Rabin–Karp dianalisis dari sisi kemampuannya menemukan *substring* identik secara cepat melalui perhitungan *rolling hash*, sedangkan Cosine Similarity dievaluasi dari sisi kemampuannya mengukur tingkat kemiripan semantik melalui representasi vektor teks.



Gambar 3.1 Hasil Deteksi Plagiarisme Dengan Metode Rabin-Karp



Gambar 3.2 Hasil Deteksi Plagiarisme Dengan Metode Cosine Similarity

Untuk akurasi, dihitung metrik *precision*, *recall*, *f1-score*, dan tingkat kesalahan deteksi baik *false positive* maupun *false negative*, Hasil pengujian kemudian dianalisis secara komparatif antara penggunaan Rabin–Karp, Cosine Similarity, maupun kombinasi keduanya, sehingga diperoleh pemahaman mendalam mengenai *trade-off* antara kecepatan deteksi dan ketepatan hasil dalam konteks deteksi plagiarisme teks Bahasa Indonesia.

3.6. Contoh Teks Plagiarisme

Plagiarisme adalah penggunaan ide, kata – kata, atau karya orang lain tanpa pengakuan yang layak sehingga pembaca berfikir itu milik penulis yang mengutip. Dibawah ini saya berikan contoh (dengan teks sampel) yang menunjukkan

perbedaan antara teks yang plagiat dan yang bukan plagiat disini saya menggunakan website https://www.duplichecker.com/ untuk mengecek plagiarisme sebuah teks.

a. Contoh Teks yang Plagiat

Perkembangan teknologi informasi semakin berkembang pesat. Terutama di bidang smartphone. Smartphone di Indonesia kini hadir dengan harga yang semakin terjangkau, yang menyebabkan jumlah pengguna smartphone di Indonesia meningkat. Salah satu sistem operasi smartphone yang sedang berkembang saat ini adalah android. Tidak bisa dipungkiri perkembangan smartphone yang begitu pesat membuat banyak aplikasi menggunakan Augmented Reality (AR) untuk membuat aplikasi menjadi menarik. Augmented Reality (AR) sendiri merupakan teknologi yang menggabungkan objek virtual ke dalam lingkungan nyata. Augmented Reality (AR) bertujuan untuk menggabungkan media digital 3D untuk memberikan bentuk objek yang lebih spesifik di mana pengguna juga dapat berinteraksi dengan objek. Penerapan teknologi Augmented Reality (AR) banyak digunakan di bidang kesehatan, militer, arsitektur, hiburan, navigasi, pendidikan dan bidang lainnya.

Dengan bantuan teknologi Augmented Reality (AR), keseluruhan bentuk Interior Istana Maimun yang akan diajarkan dapat ditampilkan dalam bentuk 3D, dengan harapan dapat meningkatkan efektifitas dalam mengidentifikasi bermacam-macam interior istana maimun. Selain itu, pengguna dapat menerapkan Augmented Reality (AR) langsung melalui kamera smartphone android. Dalam pengenalan interior istana maimun, teknologi Augmented Reality (AR) dapat menarik perhatian para client dan anak-anak, karena pengenalan interior istana maimun hanya menggunakan gambar 2D kurang menarik bagi client dan anak-anak.

b. Contoh Teks yang tidak Plagiat atau sudah diperbaiki

Perkembangan teknologi informasi semakin berkembang pesat. Terutama di bidang smartphone. Smartphone di Indonesia kini hadir dengan harga yang semakin terjangkau, yang menyebabkan jumlah pengguna

smartphone di Indonesia meningkat. Salah satu sistem operasi smartphone yang sedang berkembang saat ini adalah android. Tidak bisa dipungkiri perkembangan smartphone yang begitu pesat membuat banyak aplikasi menggunakan stoked Reality(AR) untuk membuat aplikasi menjadi menarik. stoked Reality(AR) sendiri merupakan teknologi yang menggabungkan objek virtual ke dalam lingkungan nyata. stoked Reality(AR) bertujuan untuk menggabungkan media digital 3D untuk memberikan bentuk objek yang lebih spesifik di mana pengguna juga dapat berinteraksi dengan objek. Penerapan teknologi stoked Reality(AR) banyak digunakan di bidang kesehatan, militer, arsitektur, hiburan, navigasi, pendidikan dan bidang lainnya.

Dengan bantuan teknologi stoked Reality(AR), keseluruhan bentuk Interior Istana Maimun yang akan diajarkan dapat ditampilkan dalam bentuk 3D, dengan harapan dapat meningkatkan efektifitas dalam mengidentifikasi bermacam- macam innards istana maimun. Selain itu, pengguna dapat menerapkan stoked Reality(AR) langsung melalui kamera smartphone android. Dalam pengenalan innards istana maimun, teknologi stoked Reality(AR) dapat menarik perhatian para customer dan anak- anak, karena pengenalan innards istana maimun hanya menggunakan gambar 2D kurang menarik bagi customer dan anak- anak.

Teks dengan background merah menunjukkan teks tersebut terdeteksi plagiarisme atau kalimat sama persis dengan sumber lainnya, sedangkan teks yang bewarna kuning kalimat mirip atau parafrase sebagian plagiat dan untuk teks yang tidak ada backgroundnya berarti tidak ditemukan kemiripan.

3.7.Waktu dan Tempat Penelitian

3.7.1. Waktu Penelitian

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

Tabel 3. 1 Waktu Penelitian

3.7.2. 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.

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.