

Perbandingan Kinerja Algoritma *Jaro-Winkler* dan *Levenshtein Distance* untuk Deteksi Kesalahan Penulisan Bahasa Indonesia pada Karya Ilmiah Mahasiswa

Darmanto^{*1}, Kharisma², Ar-Razy Muhammad³, Novi Indah Pradasari⁴, Eka Wahyudi⁵

^{1,2,3,4}Jurusan Teknik Elektro dan Teknik Informatika, Politeknik Negeri Ketapang,

Ketapang – Kalimantan Barat

e-mail: ^{*1}darmanto@politap.ac.id, ²kharisma@politap.ac.id, ³ar-razymuhammad@politap.ac.id,

⁴novi.ip@politap.ac.id, ⁵eka.wahyudi@politap.ac.id

Abstrak

Penelitian ini meneliti keterampilan menulis mahasiswa di Politeknik Negeri Ketapang, khususnya dalam konteks penulisan karya ilmiah. Fokus utama penelitian adalah mengidentifikasi dan memperbaiki kesalahan umum, termasuk penggunaan ejaan yang kurang tepat, pemilihan kata yang tidak sesuai, susunan kalimat yang kurang koheren, dan pembentukan paragraf yang kurang efektif. Pentingnya pemahaman tata bahasa dan kaidah kebahasaan dalam menciptakan tulisan yang runtut menjadi sorotan dalam kajian ini. Kesalahan ejaan, sebagai elemen krusial dalam penulisan, dapat dikurangi melalui penerapan aturan yang dijelaskan dalam Pedoman Umum Ejaan Bahasa Indonesia (PUEBI). Selain itu, permasalahan *typographical error* yang timbul dari tata letak huruf pada keyboard dapat diatasi dengan mempertimbangkan usulan ejaan dari Kamus Besar Bahasa Indonesia. Penelitian ini juga mengusulkan pemanfaatan metode *Approximate String Matching (ASM)*, terutama Algoritma *Jaro-Winkler* dan *Levenshtein Distance*, sebagai solusi untuk mengidentifikasi dan memberikan rekomendasi perbaikan terhadap kesalahan penulisan. Tujuan utama penelitian adalah mengembangkan sistem informasi pendeteksi kesalahan penulisan ejaan yang dapat digunakan oleh mahasiswa sebagai alat bantu untuk meningkatkan kualitas Tugas Akhir sesuai dengan standar penulisan yang berlaku. Harapannya, hasil penelitian ini dapat secara efektif meningkatkan kualitas karya tulis ilmiah mahasiswa di Politap, sekaligus memberikan kontribusi positif pada pengembangan sistem pengecekan kesalahan penulisan bahasa Indonesia secara lebih luas.

Kata kunci— *Jaro-Winkler, Levenshtein Distance, Kesalahan Penulisan, Karya Ilmiah Mahasiswa*

1. PENDAHULUAN

Pendidikan tinggi merupakan wadah bagi mahasiswa untuk menempuh pendidikan tingkat lanjut jenjang diploma, sarjana, magister dan doktor. Untuk dapat lulus dan memperoleh gelar sesuai dengan bidang keilmuannya dari perguruan tinggi maka mahasiswa harus mengambil salah satu syarat wajib yaitu menyelesaikan atau membuat sebuah karya ilmiah berupa tugas akhir, skripsi, tesis maupun disertasi tergantung jenjang yang sedang ditempuhnya. Karya ilmiah adalah tulisan yang diterbitkan dengan tujuan untuk menyampaikan hasil penelitian yang dilakukan sesuai dengan kaidah penulisan yang berlaku, sebagai bentuk karya tulis karya ilmiah disusun dan dipublikasikan berdasarkan hasil observasi, penelitian, pengamatan, serta kajian dalam bidang tertentu yang dapat dipertanggungjawabkan secara ilmiah [1]. Dalam penulisan karya ilmiah, penggunaan Bahasa Indonesia ragam baku menjadi keharusan untuk membantu pembaca memahami isi tulisan dengan lebih jelas dan mengurangi risiko terjadinya kesalahpahaman [2], selain itu juga penting untuk memperhatikan penggunaan tanda baca,

pemilihan kata, dan kesesuaian dengan Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) [3].

Kesalahan penulisan bahasa Indonesia adalah masalah umum yang sering terjadi pada karya tulis ilmiah, terutama bagi mahasiswa di Politeknik Negeri Ketapang (Politap). Hal ini dapat menyebabkan penurunan kualitas karya tulis ilmiah dan bahkan dapat mempengaruhi nilai akhir mahasiswa. Oleh karena itu, diperlukan suatu sistem pengecekan kesalahan penulisan bahasa Indonesia yang efektif untuk membantu meningkatkan kualitas karya tulis ilmiah mahasiswa. Dengan adanya aturan PUEBI, diharapkan para peneliti atau penulis karya ilmiah dapat mengurangi kesalahan dalam penulisan. Kesalahan yang sering terjadi dalam penulisan Bahasa Indonesia meliputi penggunaan huruf kapital yang tidak tepat dan kesalahan dalam pemakaian tanda baca. Kesalahan semacam ini dapat menyebabkan ambiguitas bagi pembaca dan bahkan mengubah arti serta makna suatu kalimat. Oleh karena itu, banyak peneliti terdorong untuk mencari solusi atas permasalahan tersebut melalui berbagai penelitian.

Kesalahan penulisan dokumen juga dapat terjadi diakibatkan oleh tata letak huruf pada keyboard yang berdekatan sehingga memungkinkan terjadinya kesalahan dalam penulisan ejaan atau kata. Kesalahan penulisan atau typographical error dapat dibagi menjadi beberapa jenis, yaitu: penambahan (insertion), penghapusan (deletion), dan pertukaran (substitution) suatu karakter di dalam kata atau frasa [4]. Pendeteksian kesalahan penulisan secara manual memerlukan waktu dan tenaga yang tidak sedikit, terutama jika jumlah karya ilmiah yang diperiksa cukup banyak. Oleh karena itu, diperlukan sebuah sistem otomatis yang mampu mendeteksi kesalahan penulisan secara efektif dan efisien. Salah satu pendekatan yang dapat digunakan adalah penerapan algoritma pengukuran kesamaan string untuk mendeteksi kesalahan penulisan, seperti algoritma Jaro-Winkler dan Levenshtein Distance.

Algoritma Jaro-Winkler dan Levenshtein Distance merupakan dua metode populer dalam pengolahan teks untuk mengukur kesamaan antara dua string. Algoritma Jaro-Winkler dikenal unggul dalam mendeteksi kesamaan teks dengan mempertimbangkan urutan karakter dan penalti untuk transposisi, sehingga cocok untuk mendeteksi kesalahan penulisan kecil seperti salah ketik [5]. Di sisi lain algoritma Levenshtein Distance menghitung jumlah operasi minimum seperti penyisipan, penghapusan, atau penggantian karakter yang diperlukan untuk mengubah satu string menjadi string lainnya, menjadikannya efektif dalam mengidentifikasi berbagai jenis kesalahan penulisan [6]. Namun untuk mengembangkan sistem tersebut perlu dilakukan penelitian yang membandingkan keefektifan kedua algoritma ini dalam mendeteksi kesalahan penulisan bahasa Indonesia pada karya tulis ilmiah khususnya untuk mahasiswa di Politap.

Penelitian terdahulu yang dilakukan oleh [7] dan [8] membandingkan kinerja algoritma Jaro Winkler Distance untuk menentukan mana yang paling akurat dalam mencari string dengan kesalahan penulisan, evaluasi menggunakan nilai mean average precision (MAP) menunjukkan bahwa algoritma Jaro Winkler Distance memiliki kinerja terbaik dengan MAP sebesar 0,87 pada 50 kata yang salah. Jaro Winkler menghasilkan tingkat kemiripan 76% dan waktu 0,0052 detik, sementara Levenshtein Distance memperoleh tingkat kemiripan 57% dengan waktu 0,00028 detik. Hasil analisis perbandingan menunjukkan bahwa algoritma Jaro Winkler memiliki rata-rata nilai similarity sebesar 80,92%, sementara algoritma Levenshtein Distance mencatatkan rata-rata similarity sebesar 49,43%. Dari segi waktu proses, algoritma Jaro Winkler membutuhkan rata-rata waktu 0,054 detik, sedangkan algoritma Levenshtein Distance memerlukan rata-rata waktu 0,138 detik. Berdasarkan hasil perbandingan ini, algoritma Jaro Winkler terbukti memiliki akurasi similarity yang lebih tinggi dan waktu proses yang lebih cepat dibandingkan algoritma Levenshtein Distance dalam mendeteksi plagiarisme dokumen [9].

Sementara hasil pengujian [10] mendapatkan hasil yang berbeda dimana metode Levenshtein Distance menghasilkan nilai accuracy, recall, dan F1-score yang lebih baik dibandingkan dengan Jaro-Winkler Distance. Penerapan koreksi kata menggunakan metode Levenshtein Distance juga meningkatkan nilai accuracy, recall, dan F1-score jika dibandingkan dengan kondisi tanpa koreksi kata pada tahap preprocessing. Implementasi algoritma Jaro-

Winkler banyak digunakan dalam berbagai kebutuhan seperti mengatasi permasalahan plagiarisme dan kesalahan penulisan yang sering terjadi pada karya tulis mahasiswa berbasis Natural Language Processing (NLP) dengan kombinasi algoritma TF-IDF [11]. Implementasi Algoritma Jaro-Winkler Distance digunakan untuk daftar saran kandidat perbaikan kata dan N-Gram pada sistem ini digunakan untuk mencari saran perbaikan terbaik dari daftar kandidat, hasil terbaik yang didapat sebesar 85.7% dan hasil terkecil sebesar 45% [12].

Berdasarkan penelitian yang pernah dilakukan sebelumnya dengan beberapa algoritma dan metode yang digunakan, maka penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja algoritma Jaro-Winkler dan Levenshtein Distance dalam mendeteksi kesalahan penulisan bahasa Indonesia pada karya ilmiah mahasiswa. Dengan menganalisis akurasi, efisiensi, dan relevansi hasil deteksi dari kedua algoritma tersebut, diharapkan penelitian ini dapat memberikan kontribusi dalam pengembangan sistem pendeteksian kesalahan penulisan yang lebih andal. Hasil dari penelitian ini diharapkan mampu menjadi solusi yang mendukung peningkatan kualitas karya ilmiah mahasiswa serta mempermudah proses koreksi bagi para dosen dan pembimbing.

Penggunaan teknologi dalam mendeteksi kesalahan penulisan pada karya tulis ilmiah telah banyak digunakan di berbagai negara. Namun di Indonesia masih sangat sedikit yang menggunakan teknologi untuk tujuan tersebut. Sebagai lembaga pendidikan tinggi yang mengutamakan kualitas pendidikan, Politeknik Negeri Ketapang perlu memastikan bahwa karya tulis ilmiah mahasiswa yang dihasilkan berkualitas dan bebas dari kesalahan penulisan. Oleh karena itu diharapkan penelitian ini dapat memberikan kontribusi bagi pengembangan sistem pengecekan kesalahan penulisan bahasa Indonesia pada karya tulis ilmiah mahasiswa di Politeknik Negeri Ketapang dan dapat membantu pihak kampus dalam mengembangkan sistem pengecekan kesalahan penulisan bahasa Indonesia yang lebih efektif dan efisien.

2. METODE PENELITIAN

Untuk mencapai hasil penelitian yang baik maka diperlukan pemilihan metode penelitian yang tepat. Penelitian ini menguji efektifitas kinerja antara algoritma Levenshtein Distance dan Jaro-Winkler Distance dalam proses koreksi kata pada penulisan teks berbahasa Indonesia pada karya ilmiah mahasiswa di Politeknik Negeri Ketapang. Proses penelitian dilakukan melalui beberapa langkah yang disajikan sebagai berikut.

2.1. Metode Pengumpulan Data

Metode dan instrumen pengumpulan data memegang peran penting dalam pelaksanaan penelitian untuk memastikan data yang valid dan relevan [13]. Pada tahapan pengumpulan data dilakukan menggunakan beberapa metode sesuai dengan kebutuhan pengembangan sistem, di antaranya:

- a. Studi Pustaka: dilakukan dengan mengumpulkan referensi Pustaka dan pemahaman berkaitan dengan metode Approximate String Matching yang akan digunakan sebagai metode untuk mengidentifikasi kesalahan penulisan teks.
- b. Observasi: dilakukan dengan melakukan penelusuran Tugas Akhir Mahasiswa Prodi Teknologi Informasi Jurusan Teknik Elektro dan Teknik Informatika Politeknik Negeri Ketapang dan menjadikannya sebagai dataset pada penelitian ini.

2.2. Metode Pengembangan Sistem

Pengembangan sistem dalam penelitian ini mengadopsi pendekatan *agile software development method* dengan menggunakan pendekatan model *Scrum*. *Agile software development* adalah metode pengembangan perangkat lunak yang bersifat iteratif dan terstruktur dengan solusi

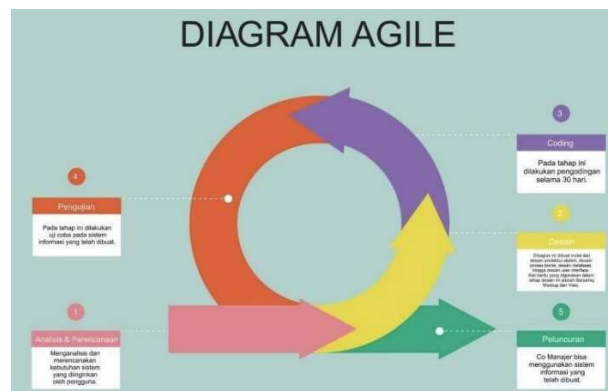
yang disepakati oleh tim secara terorganisir dan kemampuannya beradaptasi terhadap perubahan [14]. Adapun tahapan-tahapan yang dilakukan pada penelitian ini adalah sebagai berikut:

- Backlog* adalah antrian pekerjaan yang akan dilakukan dalam proyek. Tahapan penyusunan *backlog* berarti tahapan penyusunan rincian prioritas pada fitur-fitur yang akan dibangun dalam sistem. *Backlog* terdiri dari dua yaitu *Product Backlog* yang berisi seluruh daftar tugas atau daftar fitur yang akan dikerjakan dalam pengembangan sistem, dan *Sprint Backlog* yang berisi daftar tugas yang akan dikerjakan dalam suatu periode sprint tertentu.
- Sprint* adalah rangkaian kegiatan pengerjaan sistem yang terdiri dari beberapa iterasi, setiap iterasi akan dilakukan dalam waktu 2 minggu dengan target pengerjaan yang sudah ditentukan pada *Sprint Backlog*.
- Scrum Meeting* dilakukan sebagai refleksi pelaksanaan penelitian dengan membahas perkembangan penelitian setiap iterasi.
- Demo* dilakukan setiap iterasi dengan melakukan simulasi aplikasi untuk dievaluasi oleh pengguna dengan waktu yang ditentukan.

Sementara itu, di dalam setiap iterasi kegiatan pengembangan tersebut terdiri dari rangkaian kegiatan sebagai berikut:

- Analisis kebutuhan dilakukan asesmen terhadap pengguna untuk menggali secara detail *software requiremen* yang dibutuhkan oleh pengguna.
- Design* dilakukan perancangan sistem, mulai dari perancangan arsitektur sistem, desain arus data, desain database, hingga desain antar muka pengguna.
- Code Generation* dilakukan penulisan kode program menggunakan Bahasa Pemrograman PHP dan MySQL sebagai software untuk manajemen database.
- Testing untuk meminimalkan kesalahan yang terjadi pada tahapan sebelumnya sehingga memastikan luaran produk telah sesuai dengan user requirement.

Tahapan-tahapan pada *Agile Software Development Method* terdiri dari Analisis dan Perencanaan, Desain, Coding, Pengujian, dan Peluncuran. Tahapan tersebut dapat dilihat pada gambar 1 berikut.



Gambar 1. Diagram Alir *Agile Software Development*
3. HASIL DAN PEMBAHASAN

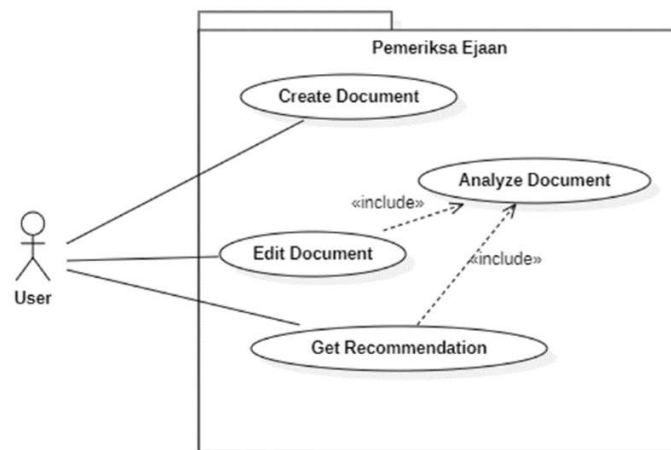
Penelitian ini menghasilkan luaran berupa sistem informasi berbasis website yang dapat melakukan pengecekan kesalahan penulisan teks dalam Bahasa Indonesia pada karya tulis ilmiah atau tugas akhir mahasiswa Politeknik Negeri Ketapang. Dataset yang digunakan dalam penelitian ini terdiri dari 10 dokumen Tugas Akhir Mahasiswa Jurusan Teknik Elektro dan Teknik

Informatika pada tahun 2022 yang diambil secara acak, adapun data yang dikumpulkan adalah teks yang terdapat pada bagian Latar Belakang pada tugas akhir mahasiswa. Tabel 1 berikut menunjukkan dataset yang digunakan pada penelitian ini.

Tabel 1. Dataset Tugas Akhir Mahasiswa

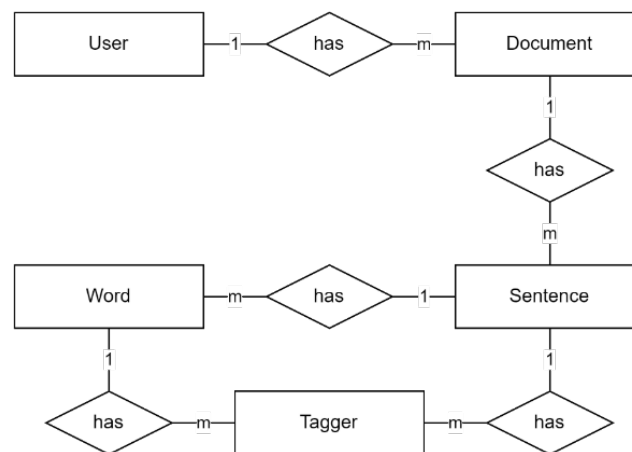
No	Nama Dokumen	Jumlah Paragraf	Jumlah Kalimat	Jumlah Kata
1.	Dokumen Tugas Akhir 1	5	10	316
2.	Dokumen Tugas Akhir 2	6	21	451
3.	Dokumen Tugas Akhir 3	4	9	234
4.	Dokumen Tugas Akhir 4	7	22	597
5.	Dokumen Tugas Akhir 5	8	21	605
6.	Dokumen Tugas Akhir 6	5	19	365
7.	Dokumen Tugas Akhir 7	7	22	621
8.	Dokumen Tugas Akhir 8	5	14	598
9.	Dokumen Tugas Akhir 9	5	10	495
10.	Dokumen Tugas Akhir 10	6	16	540
Total		58	164	4822
Rata-rata		5.8	16.4	482.2

Sementara itu di dalam setiap iterasi kegiatan pengembangan sistem terdiri dari rangkaian kegiatan Analisis Kebutuhan untuk asesmen terhadap pengguna untuk menggali secara detail *software requiremen* yang dibutuhkan dan design perancangan sistem yang digambarkan menggunakan UML (*Unified Modeling Language*) dengan usecase diagram seperti pada gambar 2 berikut. Use case merupakan bagian awal dari UML yang berperan sebagai artifak kunci dalam menentukan kesuksesan proses pengembangan perangkat lunak [15].



Gambar 2. Use Case Diagram

Selain itu dilakukan juga desain basis data menggunakan Entity Relationship Diagram (ERD) yang nantinya akan menjadi dasar dalam membuat database [16], adapun ERD pada penelitian ini dapat dilihat pada gambar 3.

Gambar 3. *Entity Relationship Diagram (ERD)*

Dalam penelitian ini, beberapa skenario pengujian digunakan untuk membandingkan Algoritma Jaro-Winkler dan Algoritma Levenshtein Distance dalam konteks deteksi kemiripan atau perbedaan antara dua string teks. Setiap skenario dirancang untuk mengevaluasi kemampuan dan efektivitas kedua algoritma dalam situasi yang berbeda.

Skenario 1: Pengujian Kinerja Dasar

Pada skenario ini, kinerja dasar kedua algoritma akan diuji menggunakan data tes sederhana. Langkah-langkahnya meliputi penyediaan sepasang string teks yang berbeda, kemudian menerapkan Algoritma Jaro-Winkler dan mencatat tingkat kemiripan yang diperoleh. Selanjutnya Algoritma Levenshtein Distance akan diterapkan pada pasangan string yang sama untuk mencatat jarak editnya. Hasil dari kedua algoritma akan dibandingkan untuk menilai perbedaan dalam keakuratan dan ketepatan masing-masing metode. Diharapkan, hasil ini akan menunjukkan perbedaan yang signifikan dalam kemampuan deteksi kemiripan antara kedua algoritma.

Skenario 2: Pengujian pada String dengan Perubahan Karakter

Skenario ini bertujuan untuk memeriksa sejauh mana kedua algoritma sensitif terhadap perubahan karakter dalam string. Untuk pengujian ini, dua string teks yang mirip namun memiliki beberapa karakter yang berbeda akan disiapkan. Algoritma Jaro-Winkler dan Levenshtein Distance akan diterapkan pada pasangan string tersebut, dan hasilnya akan dicatat. Perbandingan tingkat kemiripan dan jarak edit yang dihasilkan oleh kedua algoritma akan dilakukan untuk menentukan apakah salah satu algoritma lebih sensitif terhadap perubahan karakter dibandingkan yang lain.

Skenario 3: Pengujian pada String dengan Kosakata Berbeda

Pengujian ini fokus pada kemampuan kedua algoritma dalam menangani string yang memiliki kosakata yang berbeda. Dua string teks yang mirip namun dengan kosakata yang berbeda akan disiapkan. Algoritma Jaro-Winkler dan Levenshtein Distance akan diterapkan pada pasangan string tersebut, dan hasilnya akan dicatat. Tujuan dari pengujian ini adalah untuk mengevaluasi bagaimana masing-masing algoritma menangani variasi kosakata dan seberapa baik mereka dapat mendeteksi kemiripan atau perbedaan dalam konteks kosakata yang berbeda.

Skenario 4: Pengujian pada String Panjang dan Pendek

Skenario ini dirancang untuk menguji performa kedua algoritma pada string dengan panjang yang bervariasi. Dua string, satu panjang dan satu pendek, akan disiapkan untuk pengujian. Algoritma Jaro-Winkler dan Levenshtein Distance akan diterapkan pada kedua pasangan string untuk mencatat hasil kemiripan dan jarak edit. Hasil dari

pengujian ini akan dibandingkan untuk menilai bagaimana kedua algoritma berfungsi pada string dengan panjang yang berbeda dan mengidentifikasi apakah ada perbedaan signifikan dalam performa mereka.

Skenario 5: Pengujian pada Data Multibahasa

Pengujian ini akan mengevaluasi kemampuan kedua algoritma dalam menangani teks dalam berbagai bahasa. Dua string teks dalam bahasa yang berbeda tetapi dengan kemiripan konten akan disiapkan. Algoritma Jaro-Winkler dan Levenshtein Distance akan diterapkan pada pasangan string tersebut, dan hasilnya akan dicatat. Tujuan dari pengujian ini adalah untuk menilai bagaimana kedua algoritma menangani teks multibahasa dan seberapa efektif mereka dalam mendeteksi kemiripan atau perbedaan dalam konteks bahasa yang berbeda.

Table 1 berikut merukan teks sampel dataset TA mahasiswa yang akan dilakukan pengujian dengan 5 skenario yang dijelaskan sebelumnya. Pada tabel 2 dapat dilihat dengan jumlah string yang sama menghasilkan nilai kesamaan dan waktu akurasi yang berbeda.

Tabel 2. Dataset Tugas Akhir Mahasiswa

No.	Variabel Uji	Jumlah Kata
1.	Sistem informasi manajemen merupakan alat bantu yang digunakan untuk memudahkan pengelolaan data dan informasi dalam suatu organisasi	17
2.	Sistem informasi manajemen adalah alat yang digunakan untuk memudahkan pengelolaan data serta informasi dalam sebuah organisasi	16

Berdasarkan pengujian dengan variabel uji di atas, didapat hasil Pengujian dan Analisis sebagai berikut:

Tabel 3. Hasil uji coba nilai kesamaan dan waktu eksekusi

No	Algoritma	Nilai Kesamaan	Waktu Eksekusi
1.	Jaro Winkler	0.92	0.005 detik
2.	Levenshtein Distance	0,88	0,007 detik

Tabel 3 dilakukan pengujian dengan merubah string pada tiap kalimat kemudian akan dilakukan pengujian nilai kesamaan dan waktu eksekusi dari kedua algoritma tersebut, adapun hasilnya dapat dilihat seperti pada tabel 4.

Tabel 4. Dataset Tugas Akhir Mahasiswa

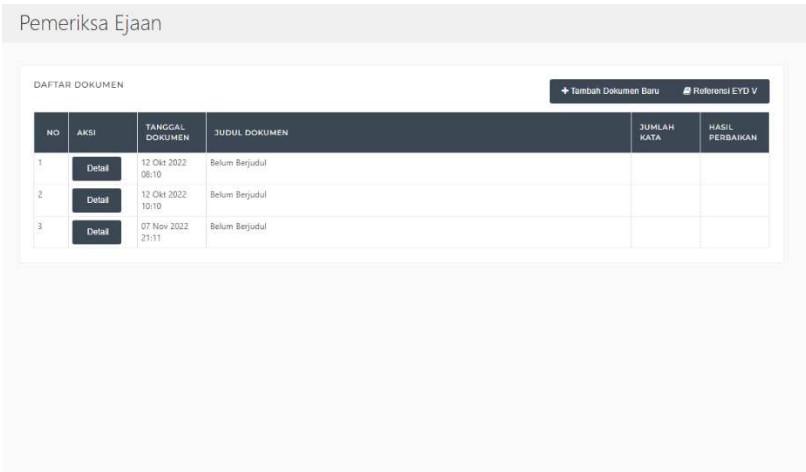
No.	Variabel Uji	Perbedaan
1.	Sistem informasi manajemen merupakan alat bantu yang digunakan untuk memudahkan pengelolaan data dan informasi dalam suatu organisasi	1. "informasi" menjadi "infromasi"
2.	Sistem informasi manajemen merupakan alat bantu yang digunakan untuk memudahkan pengelolaan data dan infromasi dalam suatu organisasi.	2. "organisasi" menjadi "organiasi"

Berdasarkan pengujian dengan variabel uji di atas, didapat hasil Pengujian dan Analisis sebagai berikut:

Tabel 5. Dataset Tugas Akhir Mahasiswa

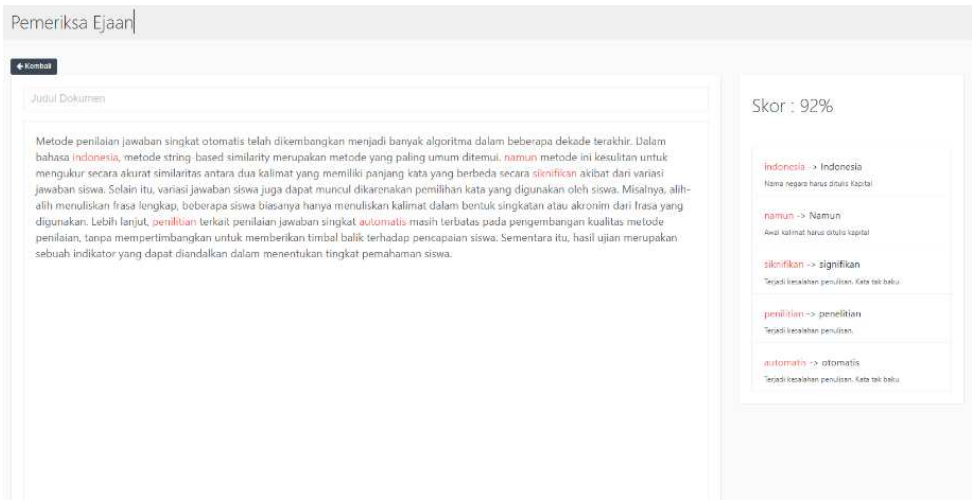
No	Algoritma	Nilai Kesamaan	Waktu Eksekusi
1.	Jaro Winkler	0.92	0.005 detik
2.	Levenshtein Distance	0,88	0,007 detik

Setelah dilakukan scenario uji kedua algoritma tersebut, tahapan berikutnya dilakukan penulisan kode program untuk memberikan perintah terhadap semua desain antarmuka aplikasi menggunakan Bahasa Pemrograman PHP dan MySQL sebagai software untuk memanajemen database. Hasil akhir pada tahapan ini adalah terbentuknya sebuah sistem infromasi yang menerapkan algortima Jaco-Winkler dan Levenshtein Distance untuk pengecekan kesalahan penulisan dalam Bahasa Indonesia pada karya tulis ilmiah berdasarkan ejaan bahasa Indonesia yang disempurnakan edisi kelima. Adapun tampilan sistem yang telah dibangun seperti pada gambar 3 berikut.



Gambar 3. Tampilan Halaman Dashboard

Tampilan dashboard merupakan halaman awal yang ditampilkan aplikasi ketika terdapat akun yang berhasil login kedalam sistem atau halaman yang menampilkan data utama dalam proses kerja sistem.



Gambar 4. Tampilan tampilan editor teks dalam Bahasa Indonesia

Pada gambar 4 Terdapat sebuah tampilan editor baru yang berfungsi untuk memasukan data berupa teks dalam bahasa Indonesia. Data teks akan terekam didalam editor kemudian di proses oleh sistem yang telah di implementasikan Algoritma Jaco-Winkler dan Levenshtein Distance sehingga dapat dilakukan pencocokan kata dan rekomendasi perbaikannya. Kolom perbaikan penulisan adalah tampilan yang menunjukan kata yang salah dan diberikan rekomendasi perbaikannya. Penulisan kata yang salah akan diberikan tanda berwarna merah dan rekomendasi perbaikan diberikan warna hitam. Rekomendasi penulisan berdasarkan dengan panduan Ejaan Bahasa Indonesia yang disempurnakan edisi ke lima.



Gambar 5. Referensi Ejaan Bahasa Indonesia

Gambar 5 merupakan tampilan referensi pedoman umum penulisan Bahasa Indonesia digunakan untuk mempermudah pengguna dalam pencarian kata yang akan digunakan dalam penyusunan Tugas Akhir, sehingga dengan adanya fitur dan aplikasi ini diharapkan mahasiswa akan lebih terbantu dalam proses penyusunan karya tulis ilmiah ataupun tugas akhir.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa algoritma Jaro-Winkler dan Levenshtein Distance memiliki kelebihan dan kekurangan masing-masing dalam konteks pengecekan kesalahan penulisan bahasa Indonesia pada karya tulis ilmiah mahasiswa Politeknik Negeri Ketapang. Algoritma Jaro-Winkler menunjukkan performa yang lebih unggul dalam hal kecepatan eksekusi dan keakuratan dalam mendeteksi kesalahan penulisan, terutama pada perubahan karakter dan variasi kosakata yang sering terjadi pada teks akademik. Di sisi lain algoritma Levenshtein Distance meskipun lebih lambat tetapi menunjukkan kemampuan yang baik dalam mendeteksi kesalahan terutama dalam kondisi teks yang lebih kompleks. Secara keseluruhan Jaro-Winkler lebih cocok digunakan dalam sistem pengecekan kesalahan penulisan bahasa Indonesia yang memerlukan respons cepat dan keakuratan tinggi. Sementara itu, Levenshtein Distance dapat menjadi alternatif yang baik dalam situasi di mana toleransi terhadap waktu eksekusi yang lebih lama dapat diterima, namun tetap memerlukan akurasi yang memadai.

DAFTAR PUSTAKA

- [1] A. Bakar, A. Sanova, Y. Yusnaidar, and F. Ekaputra, "Meningkatkan Keterampilan Guru SMAN 2 Sungai Penuh dalam Mempublikasikan Artikel Ilmiah ke Jurnal Bereputasi,"

- Lambung Inov. Pengab. Masy.*, vol. 7, no. 4, pp. 489–496, Dec. 2022, doi: 10.36312/linov.v7i4.866.
- [2] A. Misbullah, V. Mutiawani, and C. S. Mulyani, "Deteksi Kata Tak Baku dan Kesalahan Penulisan Kata pada Tugas Akhir Mahasiswa Menggunakan Metode Dictionary Lookup," *ComTech J.*, vol. 6, no. 2, p. 130, Oct. 2022, doi: 10.22373/cj.v6i2.15423.
 - [3] M. Ilmiah and A. Qoiriah, "Sistem Deteksi Kesalahan Tanda Baca dan Huruf Kapital pada Karya Tulis Ilmiah Berbahasa Indonesia Menggunakan Algoritma Boyer-Moore," *J. Inf. dan Anal. Comp. Syst. (JINACS)*, vol. 2, no. 3, pp. 185–193, Jan. 2021, doi: 10.26740/jinacs.v2n03.p185-193.
 - [4] A. I. Fahma, I. Cholissodin, and R. S. Perdana, "Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 53–62, Jan. 2018.
 - [5] E. J. Aritomatika, A. Sanjaya, and D. W. Widodo, "Implementasi Algoritma Jaro Winkler Distance untuk Pendeteksi Kesamaan Kata dalam Pengembangan Aplikasi English Conversation," in *Seminar Nasional Inovasi Teknologi*, Jul. 2021.
 - [6] Y. V. Via and R. Mumpuni, "Deteksi Kemiripan Dokumen Publikasi Skripsi Mahasiswa Menggunakan Algoritma Modifikasi Cosine Similarity," *J. Ilm. Elektro, Elektronika dan Telekomunikasi (JIEET)*, vol. 3, no. 2, pp. 57–61, Dec. 2019, doi: 10.26740/jieet.v3n2.p57-61.
 - [7] Y. Rochmawati and R. Kusumaningrum, "Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks," *J. Bina Inform.*, vol. 7, no. 2, Jan. 2016, doi: 10.24002/jbi.v7i2.491.
 - [8] I. Syahputra and F. Syakti, "Perbandingan Algoritma Levenshtein dan Jaro Winkler pada Sistem Informasi Pencarian Dokumen Perundang-undangan (Studi Kasus: Diskominfo Lahat)," *J. Sist. dan Manaj. Informatika (SMATIKA)*, vol. 12, no. 2, pp. 176–186, Dec. 2022, doi: 10.32664/smatika.v12i02.696.
 - [9] M. J. Tannga and S. Rahman, "Analisis Perbandingan Algoritma Levenshtein Distance dan Jaro Winkler untuk Aplikasi Deteksi Plagiarisme Dokumen Teks," *J. Teknol. Inf.*, vol. 4, no. 1, pp. 44–54, Mar. 2017.
 - [10] M. A. Nur, "Perbandingan Levenshtein Distance dan Jaro-Winkler Distance untuk Koreksi Kata dalam Preprocessing Analisis Sentimen Pengguna Twitter," *J. Elek.*, vol. 6, no. 2, p. 88, Jun. 2021, doi: 10.33772/jfe.v6i2.17751.
 - [11] D. Darmanto, N. I. Pradasari, and E. Wahyudi, "Sistem Deteksi Plagiarisme Tugas Akhir Mahasiswa Berbasis Natural Language Processing Menggunakan Algoritma Jaro-Winkler dan TF-IDF," *SmartComp*, vol. 13, no. 1, Jan. 2024, doi: 10.30591/smartcomp.v13i1.6375.
 - [12] N. C. Dewi and A. Qoiriah, "Implementasi Algoritma Jaro-Winkler Distance dan N-Gram untuk Deteksi dan Prediksi Perbaikan Kesalahan Penulisan Kata Bahasa Indonesia pada Karya Tulis Ilmiah Mahasiswa," *J. Inf. dan Anal. Comp. Syst. (JINACS)*, vol. 2, no. 3, pp. 169–177, Jan. 2021, doi: 10.26740/jinacs.v2n03.p169-177.
 - [13] M. Yasin, S. Garancang, and A. A. Hamzah, "Metode dan Instrumen Pengumpulan Data (Kualitatif dan Kuantitatif)," *J. Edukasi Penelit.*, vol. 2, no. 3, 2024.
 - [14] R. Wahyuningroem, "Pencatatan Transaksi Keuangan Berbasis Web Menggunakan Model Agile Scrum Development," *Ekonomika dan Bisnis (EB)*, vol. 29, no. 1, pp. 149–163, 2024, doi: 10.35760/eb.2024.v29i1.10376.
 - [15] T. A. Rospricilia and M. N. P. Ma'ady, "Pemodelan Integration Use Case (IUC): Perancangan Use Case Diagram (UML) untuk Sistem-sistem yang Terintegrasi," *J. Inf. Technol.*, vol. 9, no. 2, Sep. 2024, doi: 10.31284/j.integer.2024.v9i2.6345.
 - [16] P. R. Togatorop, R. P. Simanjuntak, S. B. Manurung, and M. C. Silalahi, "Pembangkit Entity Relationship Diagram dari Spesifikasi Kebutuhan Menggunakan Natural Language Processing untuk Bahasa Indonesia," *J. Inform. Comput.*, vol. 9, no. 2, pp. 196–206, Oct. 2021, doi: 10.35508/jicon.v9i2.5051.