JOURNAL DIGITAL TECHNOLOGY TREND
E-ISSN: 2963-8143, P-ISSN: 2963-9190
Volume 5 Number 2, December 2026, pp 204-218.
DOI: https://doi.org/10.56347/jdtt.v5i2.470.

Implementasi Algoritma *Levenshtein Distance* dan TF-IDF *Cosine Similarity* pada Sistem Validasi Ejaan dan Deteksi Redundansi Artikel Ilmiah Berbasis *Web*

Santi Septiyani [1*], Firamon Syakti [2], Irwansyah [3]
[1*,2] Program Studi Teknik Informatika, Sains Teknologi, Universitas Bina Darma, Kota Palembang, Provinsi Sumatera Selatan, Indonesia.
[3] Program Studi Teknik Komputer, Fakultas Vokasi, Universitas Bina Darma, Kota Palembang, Provinsi Sumatera Selatan, Indonesia.

ABSTRACT
Writing scientific articles requires attention to spelling accuracy, effective communication, and the prevention of redundancy. Manually checking for spelling errors and redundant passages is often time-consuming and requires meticulous effort. This research aims to implement the Levenshtein Distance algorithm, Term Frequency-Inverse Document Frequency (TF-IDF), and Cosine Similarity within a web-based system designed for spelling validation and redundancy detection in scientific articles. The system was developed using the Web Development Life Cycle (WDLC) methodology, encompassing planning, analysis, design, implementation, testing, and maintenance. The application accepts input in the form of raw text or .docx documents. Levenshtein Distance is employed to provide spelling correction recommendations based on a reference vocabulary, while TF-IDF and Cosine Similarity are used to calculate the degree of similarity between sections of the article. Functional testing was conducted using the black-box testing method across 15 scenarios. The test results indicate that all scenarios functioned as expected. The system integrates spelling validation and similarity analysis into a single application to facilitate a structured preliminary review of scientific articles.

ABSTRAK
Penulisan artikel ilmiah perlu memperhatikan ketepatan ejaan dan efektivitas penyampaian informasi serta menghindari redundansi. Pemeriksaan kesalahan ejaan dan redundansi secara manual membutuhkan waktu dan ketelitian yang tinggi. Penelitian ini bertujuan mengimplementasikan algoritma Levenshtein Distance, Term Frequency-Inverse Document Frequency (TF-IDF), dan Cosine Similarity pada sistem validasi ejaan dan deteksi redundansi artikel ilmiah berbasis web. Pengembangan sistem menggunakan metode Web Development Life Cycle (WDLC) yang meliputi perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Sistem menerima masukan berupa teks langsung maupun dokumen berformat .docx. Levenshtein Distance digunakan untuk memberikan rekomendasi perbaikan ejaan berdasarkan kosakata acuan, sedangkan TF-IDF dan Cosine Similarity digunakan untuk menghitung kemiripan antarbagian artikel. Pengujian fungsional dilakukan menggunakan metode *black-box testing* terhadap 15 skenario. Hasil pengujian menunjukkan bahwa seluruh skenario berjalan sesuai harapan. Sistem ini mengintegrasikan validasi ejaan dan analisis kemiripan dalam satu aplikasi untuk mendukung pemeriksaan awal artikel ilmiah secara terstruktur.

ARTICLE HISTORY
Received: 6 September 2026.
Revised: 11 September 2026.
Accepted: 10 October 2026.
Available Online: 20 October 2026.
Published: 1 December 2026.

KEYWORDS
Decision Support System; Spatial Analysis; Overlay-Buffer-Scoring; Geographic Information System; Water Permit Suspension.

KATA KUNCI
Sistem Pendukung Keputusan; Analisis Spasial; Overlay-Buffer-Scoring; Sistem Informasi Geografis; Skorsing Izin Air.

CONTACT Santi Septiyani santiseptiyani49@gmail.com Program Studi Teknik Informatika, Sains Teknologi, Universitas Bina Darma, Kota Palembang, Provinsi Sumatera Selatan, Indonesia.
© 2026 Information KAWANAD Institute

1. Pendahuluan

Perkembangan teknologi informasi mendorong peningkatan aktivitas penulisan dan publikasi artikel ilmiah oleh kalangan mahasiswa, dosen, maupun peneliti. Dalam lingkungan akademik, artikel ilmiah berfungsi sebagai media penyampaian hasil penelitian kepada pembaca sehingga kualitas penyajiannya harus diperhatikan secara saksama. Ketepatan ejaan, pemilihan kata, penyusunan kalimat, dan kejelasan informasi merupakan aspek penting yang mendukung keterbacaan artikel ilmiah. Supardi dan Armas (2024) menjelaskan bahwa artikel ilmiah berperan sebagai sarana penyampaian informasi ilmiah kepada masyarakat akademik. Sejalan dengan hal tersebut, pemahaman terhadap kaidah ejaan turut memengaruhi mutu penulisan karena kesalahan ejaan dapat mengurangi kejelasan pesan yang ingin disampaikan (Ramadhani *et al*., 2025).

Kendati demikian, berbagai permasalahan penulisan masih kerap ditemukan dalam artikel ilmiah, antara lain penggunaan kata tidak baku, salah tik (*typo*), kekeliruan penggunaan huruf kapital, serta pengulangan informasi yang tidak perlu. Iryani *et al*. (2025) menemukan adanya kesalahan ejaan pada sejumlah artikel ilmiah yang telah terindeks SINTA. Selain itu, Edeline *et al*. (2025) menyoroti redundansi dalam karya tulis ilmiah sebagai persoalan yang perlu ditangani agar penyampaian informasi tetap padat dan efektif. Permasalahan kesalahan ejaan tersebut dapat diselesaikan secara komputasional. Kusuma (2023) dalam kajian mengenai *spell checking* bahasa Indonesia menunjukkan bahwa metode berbasis jarak *string*, termasuk Levenshtein Distance, efektif digunakan untuk mendeteksi kekeliruan penulisan kata. Penelitian Suzanti *et al*. (2024) juga menerapkan Levenshtein Distance untuk koreksi kesalahan ejaan pada teks berbahasa Indonesia dan membuktikan bahwa metode tersebut mampu menghasilkan rekomendasi kata berdasarkan selisih karakter.

Levenshtein Distance bekerja dengan menghitung jumlah operasi penyisipan (*insertion*), penghapusan (*deletion*), dan penggantian (*substitution*) karakter yang diperlukan untuk mentransformasikan suatu kata menjadi kata rujukan. Pendekatan ini berguna untuk menentukan kandidat kata yang memiliki kemiripan bentuk terdekat dengan kata yang salah tik. Maulia *et al*. (2022) menerapkan Levenshtein Distance pada tahap normalisasi kata dan memperlihatkan efektivitas metode ini dalam mengukur disparitas antarkata. Santoso dan S (2024) [cek sitasi: pastikan nama lengkap penulis kedua] turut mengembangkan sistem koreksi kata tidak baku dalam dokumen bahasa Indonesia berbasis Levenshtein Distance yang menghasilkan rekomendasi perbaikan secara akurat. Pengujian koreksi ejaan bahasa Indonesia lain yang memadukan Levenshtein Distance dan *n-gram* juga mengonfirmasi keandalan pendekatan berbasis *edit distance* dalam memandu koreksi leksikal (Sari Kokong *et al*., 2024).

Di samping kesalahan ejaan, pengulangan gagasan atau informasi antarbagian artikel menuntut penanganan tersendiri karena mengurangi efisiensi penyampaian gagasan. Evaluasi kemiripan teks secara manual menjadi tidak efisien ketika dokumen memuat banyak bagian atau paragraf yang saling bersinggungan. Untuk mengatasi kendala tersebut, pendekatan pembobotan kata dapat diterapkan guna merepresentasikan isi dokumen. Metode Term Frequency-Inverse Document Frequency (TF-IDF) umum diimplementasikan untuk memetakan representasi fitur kata ke dalam vektor numerik, yang kemudian dihitung derajat kedekatan antardokumennya menggunakan Cosine Similarity. Pendekatan serupa diaplikasikan oleh Putra dan Supriana (2022) untuk mendeteksi kemiripan dokumen kode program. Nasrullah (2024) juga mengombinasikan TF-IDF dan Cosine Similarity untuk menghasilkan persentase kemiripan pada judul dokumen akademik. Meskipun demikian, Widianto *et al*. (2024) mencatat bahwa pendekatan berbasis kemunculan kata seperti TF-IDF dan Cosine Similarity tetap memiliki keterbatasan intrinsik ketika teks menggunakan leksikon yang berbeda namun bermakna sama (*sinonim*).

Pembangunan aplikasi berbasis web dipandang tepat untuk mengintegrasikan proses validasi ejaan dan analisis redundansi ke dalam satu sistem yang padu. Metode Web Development Life Cycle (WDLC) menyediakan alur pengembangan terstruktur untuk merealisasikan aplikasi web. Pratama dan Ardiansyah (2022) membuktikan penerapan siklus WDLC dalam membangun sistem berbasis web yang fungsional, sementara Rahmi *et al*. (2024) menguraikan bahwa tahapan WDLC—mulai dari perencanaan, analisis, perancangan, pengujian, implementasi, hingga pemeliharaan—memungkinkan proses rekayasa berlangsung sistematis sekaligus menjamin keterujian setiap fungsi perangkat lunak.

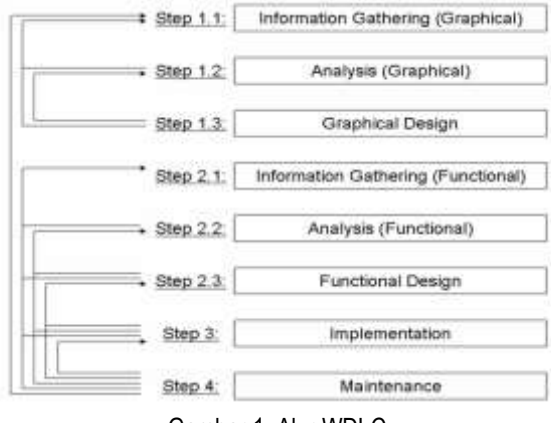
Kajian terdahulu menunjukkan bahwa Levenshtein Distance telah banyak diteliti untuk perbaikan kata tunggal, sedangkan kombinasi TF-IDF dan Cosine Similarity jamak diterapkan untuk menghitung kesamaan

dokumen. Kendati demikian, integrasi kedua ranah tersebut ke dalam satu sistem berbasis web yang berfokus pada pemeriksaan pra-publikasi artikel ilmiah masih sangat terbatas. Berangkat dari kesenjangan tersebut, penelitian ini mengintegrasikan algoritma Levenshtein Distance untuk memberikan rekomendasi koreksi ejaan serta metode TF-IDF dan Cosine Similarity untuk mengidentifikasi tingkat kemiripan atau redundansi antarbagian artikel ilmiah. Sistem ini dirancang untuk menerima berkas masukan berupa teks langsung maupun dokumen berformat *.docx*, sehingga pengguna dapat menjalankan validasi ejaan dan analisis duplikasi teks dalam satu alur kerja terpadu.

Penelitian ini bertujuan mengimplementasikan algoritma Levenshtein Distance, TF-IDF, dan Cosine Similarity ke dalam sistem validasi ejaan dan deteksi redundansi artikel ilmiah berbasis web. Di samping perancangan dan implementasi algoritma, penelitian ini juga melakukan pengujian fungsional sistem menggunakan metode *black-box testing* untuk memvalidasi seluruh fitur yang dikembangkan agar berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan.

2. Metodologi

Penelitian ini merupakan penelitian rekayasa (*Engineering Research*) yang menghasilkan artefak berupa aplikasi berbasis web untuk membantu proses validasi ejaan dan deteksi redundansi pada artikel ilmiah. Pendekatan rekayasa digunakan karena penelitian berfokus pada perancangan, pembangunan, dan evaluasi artefak sebagai solusi terhadap permasalahan yang dihadapi. Wieringa (2014) menjelaskan bahwa penelitian rekayasa perangkat lunak mencakup perancangan artefak serta investigasi terhadap artefak yang dihasilkan untuk mengetahui kesesuaiannya dengan tujuan yang ditetapkan. Di samping itu, evaluasi terhadap artefak diperlukan untuk memberikan bukti empiris mengenai kesesuaian sistem dengan kebutuhan pengguna (Revoredo *et al*., 2021). Pengembangan aplikasi dilakukan menggunakan metode Web Development Life Cycle (WDLC). Metode ini dipilih karena menyediakan tahapan pengembangan yang terstruktur untuk membangun aplikasi berbasis web, mulai dari perencanaan hingga pemeliharaan (French, 2011). Dalam penelitian ini, tahapan WDLC disesuaikan dengan kebutuhan pengembangan sistem validasi ejaan dan deteksi redundansi, yang meliputi perencanaan (*Planning*), analisis (*Analysis*), perancangan antarmuka dan sistem (*Design*), implementasi (*Implementation*), serta pemeliharaan (*Maintenance*). Alur pengembangan WDLC dalam penelitian ini disajikan pada Gambar 1.



Gambar 1. Alur WDLC

2.1 Pengumpulan Kebutuhan Sistem

Tahap pengumpulan kebutuhan (*Information Gathering*) mencakup identifikasi kebutuhan antarmuka pengguna serta kebutuhan fungsional sistem. Kebutuhan visual dikumpulkan untuk merancang tata letak, alur navigasi, skema warna, dan komponen visual antarmuka yang memudahkan interaksi pengguna. Penataan elemen visual antarmuka menjadi krusial karena berpengaruh langsung terhadap kemudahan penggunaan dan persepsi pengguna terhadap mutu perangkat lunak (Costa *et al*., 2024; Lima & Wangenheim, 2022). Sementara itu, identifikasi kebutuhan fungsional dilakukan untuk menentukan

kapabilitas yang harus disediakan oleh aplikasi (Sommerville, 2020). Kebutuhan fungsional sistem meliputi registrasi dan autentikasi pengguna, masukan artikel melalui teks langsung maupun unggah berkas .docx, validasi kesalahan ejaan, deteksi redundansi antarbagian paragraf, koreksi kata interaktif, ekspor dokumen hasil perbaikan, riwayat analisis, serta pengelolaan data pengguna. Pada tahap ini juga dikumpulkan korpus kosakata bahasa Indonesia sebagai acuan validasi kata baku. Korpus diperoleh dari repositori terbuka GitHub dan Hugging Face, yang kemudian disaring dengan membuang entri nonkata dan peribahasa sehingga menghasilkan 74.938 kata baku sebagai leksikon rujukan algoritma Levenshtein Distance.

2.2 Analisis dan Formulasi Algoritma

Tahap analisis (*Analysis*) berfokus pada pemodelan alur pemrosesan data, pemetaan struktur navigasi (Prasetya & Rahmi, 2023), serta perumusan formulasi matematis dan parameter ambang batas (*threshold*) yang digunakan untuk memproses naskah (Pressman & Maxim, 2020). Validasi ejaan dilakukan dengan membandingkan kata masukan terhadap kosakata acuan menggunakan algoritma Levenshtein Distance. Metode ini menghitung jumlah operasi penyisipan (*insertion*), penghapusan (*deletion*), dan penggantian (*substitution*) karakter minimum yang dibutuhkan untuk mentransformasikan kata masukan menjadi kata baku rujukan. Secara matematis, jarak Levenshtein D(i,j) antara prefiks kata masukan berpanjang i*i* dan kata rujukan berpanjang j*j* dinyatakan dalam Persamaan (1):

$$\mathrm{D(i,j)} = \min \begin{cases} \mathrm{D(i-1,j)} + 1 \text{ (hapus)} \\ \mathrm{D(i,j-1)} + 1 \text{ (tambah)} \\ \mathrm{D(i-1,j-1)} + cost \text{ (ganti)} \end{cases} \quad (1)$$

Dengan nilai cost=0 apabila karakter ke-i sama dengan karakter ke-j, dan cost=1 jika karakter berbeda. Kata rujukan dengan nilai jarak terendah direkomendasikan sebagai koreksi ejaan. Untuk deteksi redundansi, representasi bobot fitur kata dihitung menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF). Nilai TF mengukur frekuensi kemunculan kata dalam satu bagian, sedangkan IDF mengukur keunikan istilah pada seluruh kumpulan bagian artikel. Formulasi TF, IDF, dan TF-IDF dinyatakan berturut-turut dalam Persamaan (2), Persamaan (3), dan Persamaan (4):

$$\mathrm{TF(t,d)} = \frac{\mathrm{f(t,d)}}{\text{jumlah seluruh kata dalam dokumen}} \quad (2)$$

$$IDF(t) = log\ \frac{N}{df(t)} \quad (3)$$

$$TFIDF = TF \times IDF \quad (4)$$

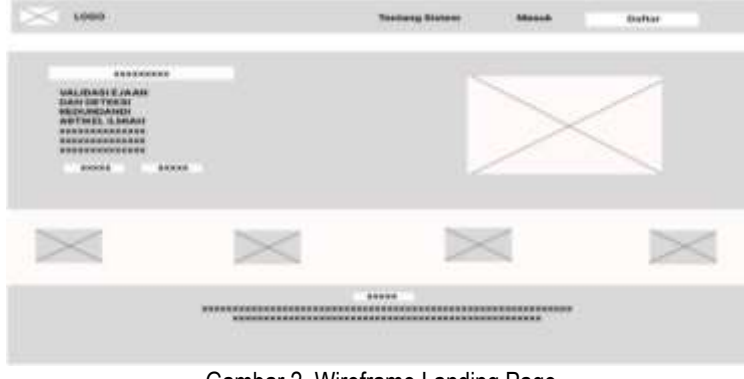
dengan f(t,d) merupakan frekuensi istilah t pada dokumen/bagian d, N adalah jumlah total bagian dokumen, dan df(t) adalah jumlah bagian yang memuat istilah t. Setelah bobot TF-IDF terbentuk dalam bentuk vektor dokumen, tingkat kemiripan antarbagian paragraf artikel dihitung menggunakan Cosine Similarity sebagaimana dirumuskan pada Persamaan (5):

$$Cosine(A,B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (5)$$

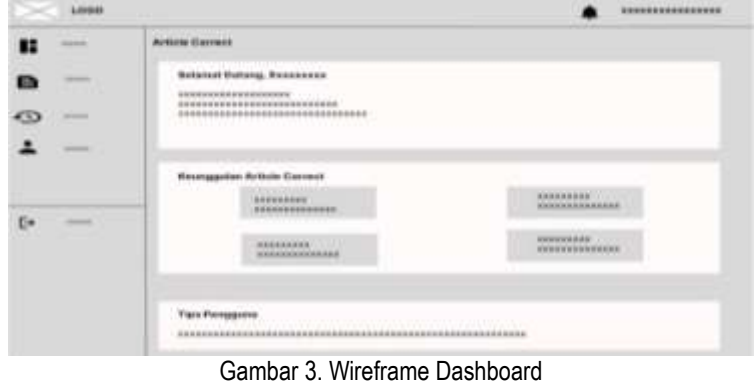
Dengan Ai dan Bi merepresentasikan bobot istilah ke-i*i* pada paragraf A dan paragraf B. Nilai Cosine Similarity berkisar antara 0 hingga 1. Pada penelitian ini ditetapkan nilai ambang batas (*threshold*) sebesar 0,80 (80%). Apabila skor kemiripan antara dua bagian paragraf mencapai atau melampaui 0,80, sistem secara otomatis menandai bagian tersebut sebagai indikasi redundansi informasi.

2.3 Perancangan Sistem dan Antarmuka

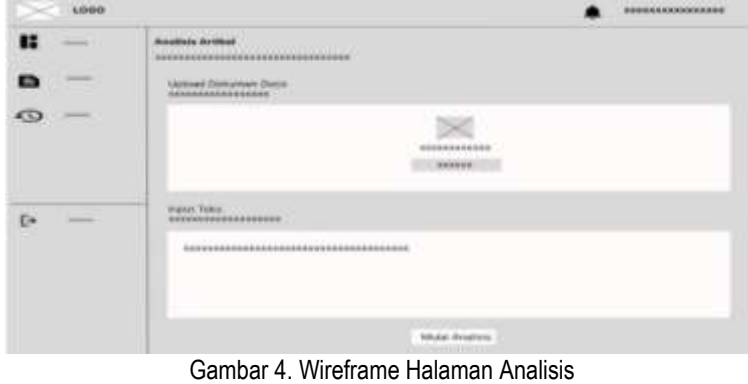
Tahap perancangan (*Design*) mencakup perancangan antarmuka pengguna dalam bentuk purwarupa visual (*wireframe*) menggunakan Figma (Lutfiyani *et al*., 2024), pemodelan fungsional menggunakan Unified Modeling Language (UML), serta perancangan basis data relasional. Rancangan antarmuka awal dituangkan dalam beberapa tampilan utama. *Wireframe Landing Page* dirancang sebagai pintu masuk awal yang menyediakan navigasi registrasi dan *login*, disajikan pada Gambar 2. Setelah proses autentikasi berhasil, pengguna diarahkan ke *Wireframe Dashboard* yang memuat ringkasan aplikasi dan petunjuk penggunaan fitur, sebagaimana diperlihatkan pada Gambar 3. Proses pemeriksaan naskah diakomodasi melalui *Wireframe Halaman Analisis* yang menyediakan opsi unggah berkas .docx maupun masukan teks langsung, sebagaimana ditunjukkan pada Gambar 4.



Gambar 2. Wireframe Landing Page



Gambar 3. Wireframe Dashboard

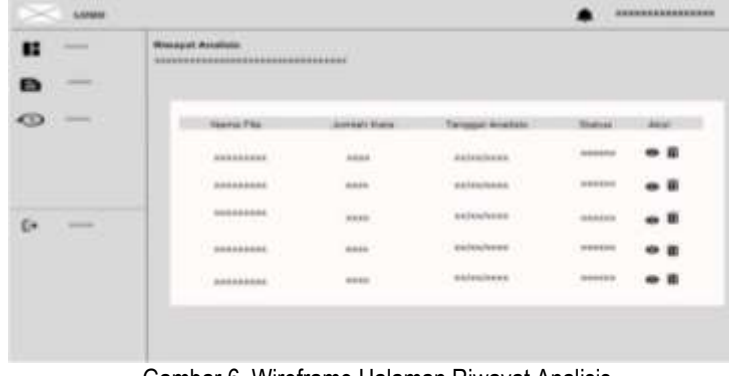


Gambar 4. Wireframe Halaman Analisis

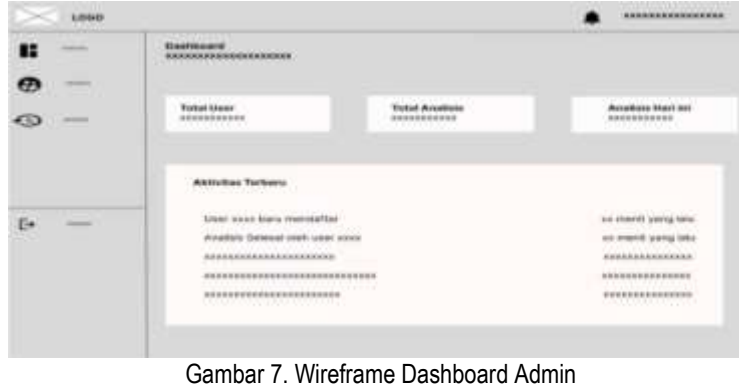
Setelah proses komputasi berlangsung, ringkasan metrik artikel, daftar kata tidak baku, dan deteksi kemiripan teks ditampilkan melalui antarmuka editor yang ditunjukkan pada Gambar 5. Riwayat pemeriksaan dokumen sebelumnya diakses melalui antarmuka pada Gambar 6, yang memungkinkan pengguna melihat ulang atau menghapus rekaman analisis. Sementara itu, antarmuka administratif untuk mengelola data akun dan statistik operasional sistem disajikan pada Gambar 7.



Gambar 5. Wireframe Halaman Hasil Analisis

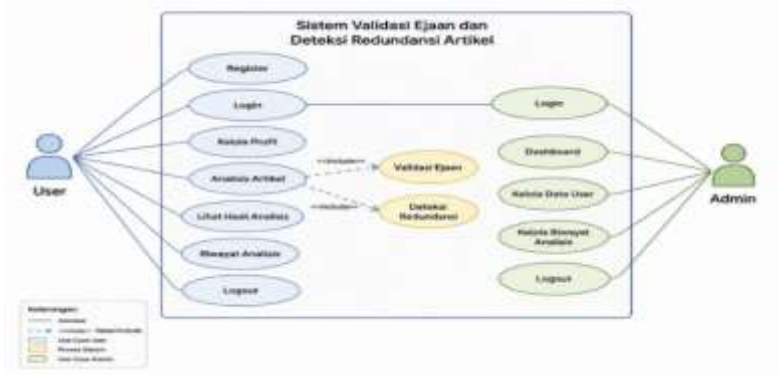


Gambar 6. Wireframe Halaman Riwayat Analisis

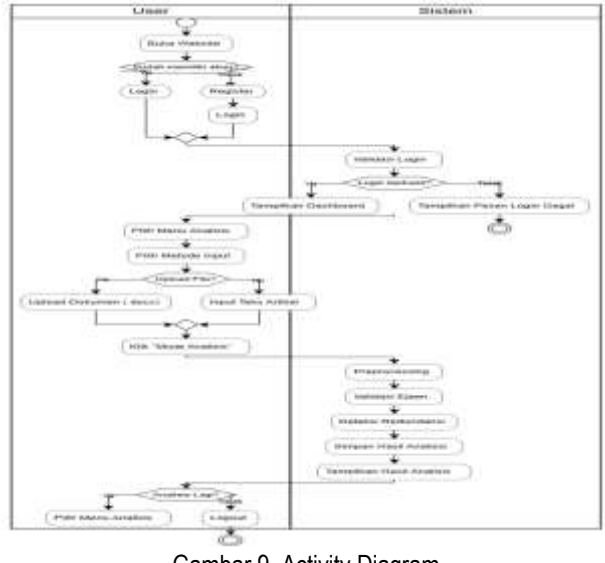


Gambar 7. Wireframe Dashboard Admin

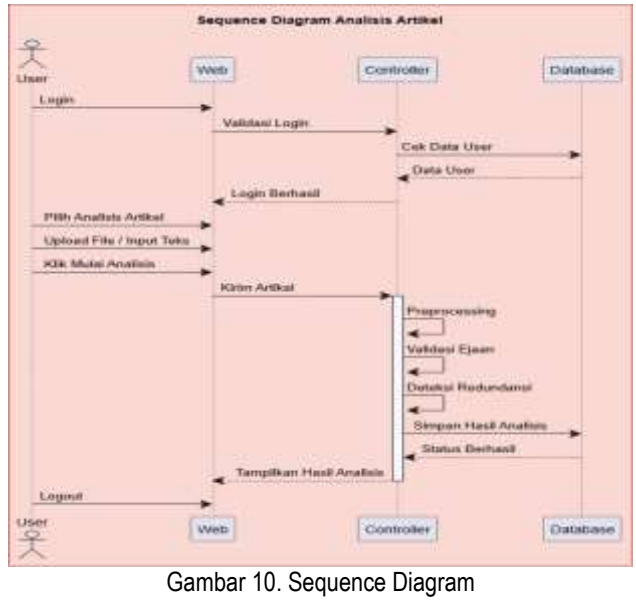
Pemodelan logika perangkat lunak dirancang menggunakan diagram UML. Relasi antara aktor (pengguna dan admin) dengan modul fungsional sistem dimodelkan melalui *Use Case Diagram* yang disajikan pada Gambar 8. Rangkaian perpindahan proses interaksi pengguna dimodelkan melalui *Activity Diagram* yang disajikan pada Gambar 9, sedangkan urutan pertukaran pesan antarkomponen perangkat lunak digambarkan melalui *Sequence Diagram* yang disajikan pada Gambar 10.



Gambar 8. Use Case Diagram



Gambar 9. Activity Diagram



Gambar 10. Sequence Diagram

Penyimpanan data sistem menggunakan sistem manajemen basis data relasional MySQL dengan nama basis data sistem_validasi_artikel. Struktur data terbagi ke dalam empat tabel utama, yaitu users, dokumen, hasil_analisis, dan detail_koreksi, yang terhubung melalui relasi *primary key* dan *foreign key*. Rancangan skema basis data disajikan melalui Entity Relationship Diagram (ERD) pada Gambar 11.



Gambar 11. Entity Relationship Diagram

2.4 Implementasi Sistem

Tahap implementasi merealisasikan rancangan arsitektur dan algoritma ke dalam aplikasi berbasis web yang fungsional. Proses komputasi validasi ejaan dijalankan dengan mencocokkan leksikon artikel terhadap dataset kosakata menggunakan Levenshtein Distance, sedangkan komputasi redundansi paragraf diterapkan melalui pembobotan TF-IDF dan Cosine Similarity (Madoa *et al*., 2026). Di samping fungsi inti tersebut, modul autentikasi pengguna, pembaca berkas dokumen berformat .docx, editor teks, modul unduh berkas hasil koreksi, dan manajemen basis data pengguna turut diintegrasikan.

2.5 Pemeliharaan Sistem

Tahap pemeliharaan (*Maintenance*) ditujukan untuk menjaga stabilitas perangkat lunak, perbaikan celah kesalahan program (*bug*), dan pembaruan berkala terhadap korpus kosakata acuan guna memperluas cakupan deteksi kata baku bahasa Indonesia.

3. Hasil

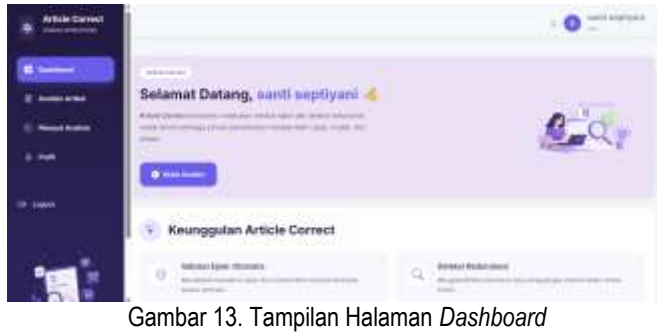
Implementasi artefak menghasilkan aplikasi berbasis web untuk validasi ejaan dan deteksi redundansi artikel ilmiah yang berfungsi sesuai spesifikasi rancangan sistem. Aplikasi ini mengintegrasikan fungsi analisis artikel secara terotomasi, penyediaan saran koreksi kata tidak baku berbasis leksikon acuan, perhitungan kemiripan paragraf untuk identifikasi redundansi, pencatatan riwayat dokumen, serta modul manajemen pengguna.

3.1 Implementasi Antarmuka Utama dan Manajemen Pengguna

Pengguna yang mengakses aplikasi pertama kali diarahkan ke antarmuka *landing page* yang berfungsi sebagai pintu masuk utama sekaligus media sosialisasi fitur sistem, sebagaimana disajikan pada Gambar 12. Melalui halaman ini, pengguna dapat mengakses tautan registrasi akun dan formulir *login*. Setelah otentikasi berhasil, sistem mengarahkan pengguna ke antarmuka *dashboard* yang menyajikan ringkasan informasi aplikasi, petunjuk operasional pengujian naskah, serta bilah navigasi utama, sebagaimana ditampilkan pada Gambar 13.



Gambar 12. Tampilan Halaman *Landing Page*



Gambar 13. Tampilan Halaman *Dashboard*

Untuk keperluan pemantauan dan pengelolaan operasional aplikasi secara menyeluruh, sistem dilengkapi antarmuka dasbor khusus administrator. Melalui antarmuka pada Gambar 14, administrator dapat memantau metrik penggunaan sistem secara langsung, meliputi total akun pengguna terdaftar, akumulasi volume artikel yang telah diproses, riwayat aktivitas terkini, serta pengelolaan hak akses akun pengguna.



Gambar 14. Tampilan Halaman *Dashboard* Admin

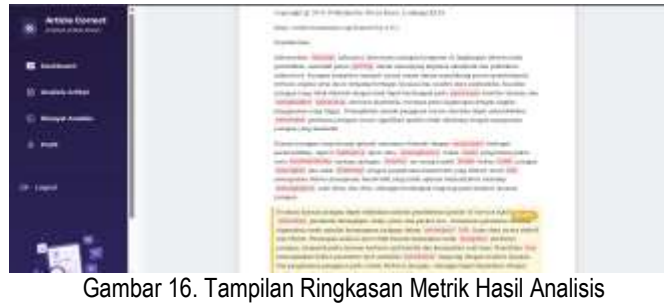
3.2 Implementasi Fitur Analisis, Editor Koreksi, dan Riwayat Dokumen

Proses pemeriksaan naskah diakomodasi melalui antarmuka halaman analisis artikel. Sistem memfasilitasi dua moda masukan data yang fleksibel, yaitu pengetikan atau penempelan teks langsung ke dalam area teks (*textarea*) serta pengunggahan berkas dokumen berformat .docx, sebagaimana diperlihatkan pada Gambar 15. Setelah berkas atau teks diunggah, penekanan tombol komputasi akan memicu proses ekstraksi teks, segmentasi paragraf, validasi leksikal menggunakan Levenshtein Distance, serta kalkulasi bobot kemiripan teks menggunakan TF-IDF dan Cosine Similarity.



Gambar 15. Tampilan Halaman Unggah Dokumen Analisis

Keluaran hasil analisis disajikan secara komprehensif pada halaman hasil pemeriksaan. Sistem menampilkan panel metrik ringkasan kuantitatif yang menginformasikan status pemrosesan, total kata, jumlah paragraf, frekuensi kesalahan ejaan, jumlah paragraf yang teridentifikasi redundan, skor kemiripan, dan progres penyelesaian perbaikan. Pada pengujian satu dokumen artikel uji berukuran 2.632 kata dan 100 paragraf, sistem berhasil mendeteksi 339 kata tidak baku serta mengidentifikasi 15 paragraf yang terindikasi redundan dengan skor tingkat redundansi sebesar 89,07%, sebagaimana ditunjukkan pada Gambar 16.



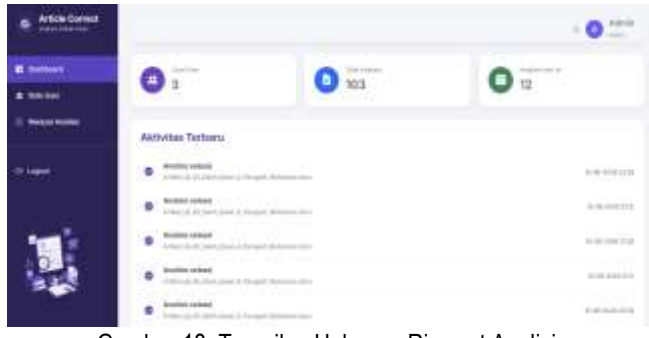
Gambar 16. Tampilan Ringkasan Metrik Hasil Analisis

Di samping ringkasan metrik, sistem menyediakan antarmuka editor interaktif untuk mempermudah perbaikan langsung pada teks naskah, sebagaimana disajikan pada Gambar 17. Kata yang teridentifikasi salah eja ditandai dengan garis bawah berwarna merah yang interaktif, di mana pengguna cukup mengeklik kata tersebut untuk memilih daftar rekomendasi kata baku dari leksikon rujukan atau memanfaatkan opsi koreksi otomatis secara menyeluruh (*Fix All*). Paragraf yang memiliki derajat kemiripan di atas ambang batas 0,80 ditandai secara visual agar pengguna dapat segera merevisi teks yang berulang.



Gambar 17. Tampilan Editor Interaktif Koreksi Artikel

Seluruh aktivitas evaluasi naskah dicatat ke dalam basis data dan dapat ditinjau kembali melalui antarmuka riwayat analisis, sebagaimana ditunjukkan pada Gambar 18. Pada halaman ini, pengguna dapat meninjau daftar naskah yang pernah diproses beserta metadata berkas, tanggal pengujian, dan ringkasan status analisis, serta membuka kembali hasil evaluasi terdahulu ke dalam lembar editor tanpa harus memproses ulang dokumen dari awal.



Gambar 18. Tampilan Halaman Riwayat Analisis

4. Pembahasan

Evaluasi fungsional sistem dilakukan menggunakan metode pengujian kotak hitam (*black-box testing*) melalui 15 skenario uji untuk memastikan setiap fitur berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Lingkup pengujian mencakup aspek otentikasi akun, pembatasan hak akses (*route guard*), fleksibilitas masukan naskah (area teks dan berkas .docx), validasi format berkas, efektivitas deteksi ejaan dan redundansi berbasis algoritma, manipulasi teks pada penyunting interaktif, ekspor berkas hasil perbaikan, hingga modul manajemen pengguna pada sisi administrator. Rincian hasil pengujian fungsional disajikan pada Tabel 1.

Tabel 1. Hasil Pengujian Kotak Hitam (*Black-Box Testing*) Sistem

Skenario Pengujian	Kasus Uji dan Masukan Data	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Registrasi Akun	Mengisi formulir pendaftaran dengan nama, *username*, *email*, dan kata sandi yang valid.	Data akun tersimpan ke basis data dan sistem mengarahkan pengguna ke halaman *login*.	Sesuai Harapan	Valid
Login Pengguna	Memasukkan *username* dan kata sandi yang telah terdaftar.	Sistem berhasil memvalidasi akun dan mengarahkan pengguna atau administrator ke dasbor masing-masing.	Sesuai Harapan	Valid
Validasi *Login* Gagal	Memasukkan *username* atau kata sandi yang tidak terdaftar/salah.	Sistem menolak permintaan masuk dan menampilkan notifikasi peringatan kegagalan autentikasi.	Sesuai Harapan	Valid
Hak Akses (*Route Guard*)	Mengakses tautan URL halaman analisis atau dasbor tanpa melakukan *login* terlebih dahulu.	Sistem memblokir akses secara langsung dan mengalihkan peramban secara otomatis ke halaman *login*.	Sesuai Harapan	Valid
Analisis via Masukan Teks	Menempelkan (*paste*) atau mengetik naskah artikel ilmiah pada area teks (*textarea*) lalu menekan tombol analisis.	Sistem mengekstrak teks, menghitung volume kata, serta menyajikan hasil analisis ke halaman editor.	Sesuai Harapan	Valid
Analisis via	Mengunggah berkas	Sistem membaca struktur	Sesuai	Valid

Unggah DOCX	dokumen artikel berformat .docx.	isi berkas, mengekstrak paragraf, dan menampilkan hasil analisis.	Harapan	
Validasi Format Berkas	Mengunggah berkas dengan format di luar ekstensi .docx (seperti .pdf atau .txt).	Sistem menolak pemrosesan berkas dan memunculkan notifikasi peringatan: "Format file harus .docx".	Sesuai Harapan	Valid
Deteksi Kesalahan Ejaan	Menguji naskah yang memuat leksikon nonbaku atau salah tik (misalnya: *analisa*, *sisti m*).	Sistem menandai kata nonbaku dengan garis bawah merah dan menyajikan rekomendasi kata baku dari kosakata acuan berbasis Levenshtein Distance.	Sesuai Harapan	Valid
Deteksi Redundansi Paragraf	Memasukkan naskah artikel yang memuat pengulangan substansi antarparagraf.	Sistem mendeteksi derajat kemiripan berbasis Cosine Similarity, menampilkan label persentase kemiripan, serta memberikan saran perbaikan.	Sesuai Harapan	Valid
Koreksi Kata pada Editor	Mengeklik kata yang bergaris bawah merah dan memilih opsi leksikon perbaikan yang direkomendasikan.	Kata pada editor teks berganti secara otomatis dengan kata baku yang dipilih dan penanda merah terhapus.	Sesuai Harapan	Valid
Fitur Perbaiki Semua (*Fix All*)	Menekan tombol "Perbaiki Semua" pada bilah alat (*toolbar*) editor.	Seluruh kata salah tik di dalam dokumen terkoreksi secara serentak berdasarkan rekomendasi leksikon teratas.	Sesuai Harapan	Valid
Fitur *Undo* dan *Re do*	Menekan tombol interaktif *Undo* atau *Re do* setelah melakukan tindakan koreksi kata.	Tindakan perbaikan kata sebelumnya berhasil dibatalkan (*undo*) atau diterapkan kembali (*redo*) dengan benar.	Sesuai Harapan	Valid
Ekspor Dokumen	Mengunduh hasil perbaikan artikel ke dalam format .docx dokumen bersih.	Sistem mengunduh berkas Microsoft Word (.docx) hasil perbaikan dengan tata format baku (Times New Roman 12 pt, spasi 1,5).	Sesuai Harapan	Valid
Riwayat Analisis Pengguna	Membuka menu riwayat untuk melihat arsip analisis terdahulu dan membuka kembali salah satu dokumen.	Daftar rekaman riwayat ditampilkan lengkap dan berkas lama dapat dimuat kembali ke lembar editor interaktif.	Sesuai Harapan	Valid
Manajemen Pengguna (Admin)	Administrator melakukan penambahan, penyuntingan data,	Data akun pengguna pada sistem berhasil diperbarui di dalam basis data sesuai dengan operasi	Sesuai Harapan	Valid

	perubahan peran (*role*), dan penghapusan akun pengguna.	administratif yang dilakukan.	

Berdasarkan data pada Tabel 1, seluruh 15 skenario pengujian fungsional memperoleh hasil yang sesuai dengan kriteria penerimaan sistem dengan kesimpulan valid, yang merepresentasikan tingkat keberhasilan fungsional sebesar 100%. Hasil ini mengonfirmasi bahwa setiap modul perangkat lunak dapat merespons berbagai variasi masukan data dan aksi pengguna tanpa menghasilkan galat eksekusi pada lingkungan peramban web. Pengujian tersebut membuktikan kelancaran alur operasional sistem dalam memfasilitasi pemeriksaan artikel dari tahap awal hingga tahap pascapemrosesan. Pengguna memperoleh kemudahan fleksibilitas melalui penyediaan dua saluran masukan, baik berupa teks lewah maupun dokumen utuh berekstensi .docx. Keberadaan mekanisme penyaring format berkas menjamin kestabilan modul pembaca dokumen dari potensi kesalahan pembacaan struktur berkas yang tidak kompatibel. Selain itu, fitur pendukung editor seperti fungsi pembatalan bertingkat (*undo/redo*) dan tombol perbaikan massal (*Fix All*) terbukti efektif mempersingkat durasi penyuntingan naskah yang memiliki kepadatan kesalahan ejaan tinggi. Pada implementasi algoritma komputasi teks, hasil pengujian menegaskan bahwa algoritma Levenshtein Distance berhasil menjalankan fungsinya dalam mendeteksi disparitas karakter leksikal dan memberikan rekomendasi kata pengganti yang relevan dari repositori kosakata acuan. Secara bersamaan, integrasi pembobotan TF-IDF dan kalkulasi Cosine Similarity terbukti mampu mendeteksi tingkat kemiripan tekstual antarbagian paragraf serta memberikan indikasi visual yang jelas terhadap bagian naskah yang berulang. Dengan berjalannya seluruh fungsi ini dalam satu platform terpadu, aplikasi yang dikembangkan berhasil memangkas fragmentasi proses pemeriksaan naskah yang sebelumnya harus dilakukan secara terpisah melalui berbagai perangkat bantu yang berbeda.

5. Kesimpulan

Sistem validasi ejaan dan deteksi redundansi artikel ilmiah berbasis web telah berhasil dibangun dan diimplementasikan menggunakan metodologi Web Development Life Cycle (WDLC). Integrasi algoritma Levenshtein Distance dengan leksikon acuan terbukti efektif memberikan rekomendasi kata baku pada kesalahan salah tik, sementara kombinasi TF-IDF dan Cosine Similarity dengan ambang batas 0,80 mampu mengidentifikasi bagian paragraf yang memuat pengulangan gagasan. Hasil pengujian fungsional kotak hitam (*black-box testing*) terhadap 15 skenario uji menunjukkan tingkat keberhasilan sebesar 100%, yang mengonfirmasi bahwa seluruh fitur—mulai dari penerimaan berkas .docx dan area teks, penyuntingan interaktif, hingga ekspor dokumen hasil koreksi—beroperasi sesuai spesifikasi yang dirancang. Keberadaan sistem ini memberikan solusi terpadu dalam memangkas fragmentasi tahapan pra-penyuntingan artikel ilmiah ke dalam satu aplikasi berbasis web yang praktis dan terstruktur. Sebagai pengembangan berkelanjutan, cakupan korpus kosakata acuan bahasa Indonesia dapat diperluas secara berkala, serta integrasi pendekatan kesamaan semantik (*semantic similarity*) berbasis model bahasa dapat diterapkan untuk mengatasi keterbatasan deteksi redundansi pada kalimat berparafrasa atau bersinonim.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Universitas Bina Darma yang telah menjadi tempat penulis menempuh pendidikan dan memperoleh berbagai ilmu serta pengalaman selama masa perkuliahan. Ucapan terima kasih yang sebesar-besarnya juga penulis sampaikan kepada dosen pembimbing yang telah dengan sabar memberikan arahan, bimbingan, masukan, dan motivasi kepada penulis sejak awal penyusunan skripsi hingga proses penyusunan dan pembuatan jurnal ini.(Supardi & Armas, 2024).

Referensi

Costa, A. N., Silva, F. O. da, & Moreira, J. J. (2024). Towards an AI-driven user interface design for web applications. *Procedia Computer Science*, *237*, 179–186. https://doi.org/10.1016/j.procs.2024.05.094

Edeline, N. P., Devi, R. S., Sintawati, K. S., & Pratama, J. B. (2025). Redundansi dalam karya tulis ilmiah: Analisis pola, dampak, dan strategi penulisan yang efektif. *Jurnal Pengembangan dan Penelitian Pendidikan*, *7*(2), 112–123.

French, A. M. (2011). Web development life cycle: A new methodology for developing web applications. *Journal of Internet Banking and Commerce*, *16*(2), 1–13.

Iryani, E., Wardani, P., Wahyuni, P., Suryani, T., & Triyanto, A. (2025). Studi berbasis korpus digital terhadap kesalahan ejaan artikel terindeks SINTA. *MABASAN*, *19*(2), 289–304. https://doi.org/10.26499/mab.v19i2.890

Kusuma, A. T. A. (2023). Spell checking bahasa Indonesia dengan melakukan perbandingan metode Peter Norvig, Levenshtein distance, dan Damerau-Lavenshtein distance: Tinjauan literatur. *AUTOMATA*, *4*(2), 1–6. https://journal.uii.ac.id/AUTOMATA/article/view/28633

Lima, A. L. de S., & Wangenheim, C. G. von. (2022). Assessing the visual esthetics of user interfaces: A ten-year systematic mapping. *International Journal of Human-Computer Interaction*, *38*(2), 144–164. https://doi.org/10.1080/10447318.2021.1926118

Lutfiyani, A., Sumantri, R. J., & Anggoro, T. (2024). User interface design techniques with wireframing for online library website layout design using Figma. *Jurnal Kridatama Sains dan Teknologi*, *6*(1), 146–159. https://doi.org/10.53863/kst.v6i01.1092

Madoa, S., Mulyadi, I., & Darniati. (2026). Implementasi metode TF-IDF dan Cosine Similarity pada sistem pencarian artikel yang relevan. *Journal of Muhammadiyah's Application Technology*, *5*(2), 85–94. https://doi.org/10.26618/a6wkdt68

Maulia, S., Rouza, E., Fimawahib, L., Bakti, I. R., & Mustafa, S. R. (2022). Normalisasi kata Bahasa Batak Mandailing menggunakan metode Levenshtein Distance. *JURSIMA (Jurnal Sistem Informasi dan Manajemen)*, *10*(3), 171–174. https://doi.org/10.47024/js.v10i3.514

Nasrullah, A. H. (2024). Integrasi TF-IDF dan algoritma Cosine Similarity untuk deteksi tingkat kemiripan judul penelitian (studi kasus mahasiswa Fakultas Ilmu Komputer UNISAN Gorontalo). *Information Technology Education Journal*, *3*(3), 113–118.

Prasetya, Y., & Rahmi, R. (2023). Pilot study on user experience analysis of Universitas Indonesia Library website. *Tik Ilmeu: Jurnal Ilmu Perpustakaan dan Informasi*, *7*(1), 105–116. https://doi.org/10.29240/tik.v7i1.6453

Pratama, J., & Ardiansyah, M. (2022). Perancangan dan pengembangan e-commerce busana menggunakan model WDLC dengan metode Scrum. *Jurnal Ilmu Komputer dan Bisnis*, *13*(2a), 120–131. https://doi.org/10.47927/jikb.v13i2a.402

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Putra, I. G. A. E., & Supriana, I. W. (2022). Deteksi plagiarisme source code tugas mahasiswa menggunakan algoritma Cosine Similarity dan pembobotan TF-IDF. *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, *1*(1), 575–582.

Rahmi, L., Asoka, E., Sulistiyanto, & Kunio, N. I. H. (2024). Analisis perancangan dan pembuatan profile website pada Enings Production menggunakan metode Web Development Life Cycle (WDLC). *Jurnal Teknologi Informatika dan Komputer*, *9*(2), 241–252. https://doi.org/10.37012/jtik.v9i2.1859

Ramadhani, R., Putri, Y. D., Musarofah, S., Sota, M. P., Lubis, N. R., Sisra, I., & Yuniati, I. (2025). Pengaruh pemahaman PUEBI terhadap ketepatan penulisan ejaan dalam karya ilmiah mahasiswa. *Jurnal Ilmiah Literasi Indonesia*, *1*(2), 400–408.

Revoredo, K., Djurica, D., & Mendling, J. (2021). A study into the practice of reporting software engineering experiments. *Empirical Software Engineering*, *26*, Artikel 113. https://doi.org/10.1007/s10664-021-10007-3

Santoso, J. T., & Setiawan, Y. (2024). A hybrid approach to typo correction in Indonesian documents using Levenshtein distance. *Journal of Technology Informatics and Engineering*, *3*(2), 151–168. https://jtie.stekom.ac.id/index.php/jtie/article/view/184

Sari Kokong, D. A. R., Irmawati, B., & Dwiyansaputra, R. (2024). Spelling error correction in Indonesian using Damerau-Levenshtein distance dan N-gram. *Jurnal Teknologi Informasi Komputer dan Aplikasinya (JTIKA)*, *6*(1), 257–263.

Sommerville, I. (2020). *Engineering software products: An introduction to modern software engineering*. Pearson.

Supardi, R., & Armas, A. M. (2024). *Metode penulisan karya ilmiah*. CV. Ruang Tentor.

Suzanti, I. O., Negara, Y. D. P., Radiah, A., Cahyani, A. D., & Husni, H. (2024). Koreksi kesalahan ejaan pada pencarian artikel pariwisata berbahasa Indonesia menggunakan Levenshtein distance. *Jurnal Simantec*, *13*(1), 81–90. https://journal.trunojoyo.ac.id/simantec/article/view/28186

Widianto, A., Pebriyanto, E., Fitriyanti, & Marna. (2024). Document similarity using term frequency-inverse document frequency representation and cosine similarity. *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, *4*(2), 149–153. https://doi.org/10.20895/dinda.v4i2.1589

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.