

PENGEMBANGAN APLIKASI PEMERIKSA TATA BAHASA INGGRIS BERBASIS WEB MENGGUNAKAN FLASK DAN INTEGRASI ARTIFICIAL INTELLIGENCE

Febe Gracia Sembiring^{1*}, Vivielda Farmawaty Tambunan², Steven Adventino Gulo³, Ayman Human Sukma⁴, Adidtya Perdana⁵

^{1,2,3,4,5}Universitas Negeri Medan; Jl. Willièm Iskandar Psr. V, Medan Estate, Medan 20221; Tel. 061-6613365, Fax. 061-6614002, 6613319

Keywords:

grammar; AI; NLP; framework Flask; web.

Correspondent Email:

febegracia45@gmail.com



JITET is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Abstrak. Kemampuan menulis bahasa Inggris yang baik menjadi tuntutan penting dalam dunia akademik dan profesional. Namun, banyak penutur non-native mengalami kesulitan dalam tata bahasa, yang dapat memengaruhi kejelasan tulisan. Penelitian ini bertujuan untuk mengembangkan aplikasi pemeriksa tata bahasa Inggris berbasis web dengan mengintegrasikan teknologi Artificial Intelligence (AI) menggunakan framework Flask dan library LanguageTool. Metode pengembangan menggunakan model Waterfall, mencakup analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Aplikasi ini mampu menerima input berupa teks langsung maupun file (.txt, .pdf, .docx), menganalisis struktur kalimat, memberikan saran perbaikan, serta menampilkan skor tata bahasa secara otomatis. Pengujian dilakukan dengan metode Black Box dan menunjukkan seluruh fitur berjalan dengan baik tanpa kesalahan mayor. Hasil penelitian menunjukkan bahwa sistem memberikan koreksi grammar secara akurat dan responsif, serta memiliki antarmuka yang ramah pengguna. Temuan ini menunjukkan potensi besar dalam mendukung pembelajaran mandiri bahasa Inggris, khususnya dalam keterampilan menulis.

Abstract. The ability to write correct English grammar is a critical requirement in academic and professional settings. However, many non-native speakers struggle with grammar, which can affect the clarity of their writing. This research aims to develop a web-based English grammar checker by integrating Artificial Intelligence (AI) technology using the Flask framework and the LanguageTool library. The system was developed using the Waterfall model, encompassing requirement analysis, system design, implementation, testing, and maintenance. The application supports both direct text input and file uploads (.txt, .pdf, .docx), analyzes sentence structure, provides correction suggestions, and automatically calculates grammar scores. Testing with the Black Box method shows that all features function correctly without major errors. The results indicate that the system provides accurate and responsive grammar feedback with a user-friendly interface. These findings highlight the potential of the application to support independent English learning, particularly in writing skills.

1. PENDAHULUAN

Kemampuan menulis dalam bahasa Inggris yang baik dan benar menjadi keterampilan penting di era globalisasi,

terutama dalam ranah akademik, profesional, dan komunikasi digital. Salah satu aspek fundamental dalam menulis adalah tata bahasa (grammar), yang berfungsi untuk

menyampaikan makna secara akurat dan terstruktur[1]. Namun, keterampilan ini tidak mudah dimiliki oleh semua orang, terutama oleh pelajar dan penutur asing. Banyak penulis mengalami kesulitan dalam menyusun kalimat yang sesuai kaidah, baik karena kurangnya pemahaman tata bahasa maupun karena kompleksitas proses menulis itu sendiri[2][3].

Proses koreksi grammar secara manual membutuhkan pemahaman linguistik yang baik, serta memerlukan waktu dan ketelitian yang tinggi. Oleh karena itu, dibutuhkan alat bantu yang mampu memberikan umpan balik grammar secara otomatis dan akurat. Dalam beberapa tahun terakhir, telah muncul berbagai aplikasi grammar checker seperti Grammarly, Hemingway, dan LanguageTool yang memanfaatkan teknologi kecerdasan buatan (Artificial Intelligence/AI) untuk mendeteksi dan memperbaiki kesalahan penulisan secara langsung[4].

Dengan berkembangnya teknologi AI dan Natural Language Processing (NLP), kini dimungkinkan untuk membangun aplikasi digital yang dapat memproses bahasa manusia secara otomatis. NLP memungkinkan sistem mengenali pola kalimat, memahami makna, dan memberikan saran perbaikan grammar[5][6]. Oleh sebab itu, penelitian ini bertujuan untuk mengembangkan aplikasi pemeriksa tata bahasa Inggris berbasis web dengan mengintegrasikan teknologi AI menggunakan pustaka LanguageTool dan framework Flask.

Aplikasi ini memungkinkan pengguna untuk memasukkan teks melalui antarmuka web maupun unggahan file (format .txt, .pdf, .docx), lalu sistem akan melakukan analisis grammar secara otomatis, memberikan saran perbaikan, serta menampilkan skor kualitas tulisan. Kontribusi utama dari penelitian ini adalah menghadirkan solusi yang bersifat praktis, mudah diakses secara daring, serta mendukung pembelajaran mandiri menulis bahasa Inggris[7][8].

Integrasi teknologi AI, web framework Flask, dan pustaka NLP menjadikan aplikasi ini sebagai sarana interaktif dan edukatif dalam meningkatkan keterampilan menulis. Dengan dukungan sistem backend ringan dan fleksibel seperti Flask, aplikasi ini dapat dengan mudah diimplementasikan dan dikembangkan lebih lanjut sesuai kebutuhan pengguna[5][9]. Inovasi ini diharapkan dapat membantu pelajar,

mahasiswa, dan profesional dalam menyempurnakan tulisan berbahasa Inggris secara efektif, efisien, dan mandiri.

2. TINJAUAN PUSTAKA

2.1 Grammar Checker Tools

Berbagai alat grammar checker berbasis AI telah dikembangkan dan digunakan secara luas, di antaranya Grammarly, Hemingway Editor, dan LanguageTool. Grammarly menggunakan teknologi AI untuk mendeteksi kesalahan grammar, ejaan, struktur kalimat, dan gaya bahasa, serta memberikan umpan balik secara otomatis[5]. Penelitian menunjukkan bahwa penggunaan Grammarly dalam pembelajaran akademik dapat meningkatkan kualitas tulisan serta memberikan umpan balik korektif yang bermanfaat[1]. Namun, terdapat kekhawatiran bahwa koreksi otomatis tidak selalu mempertimbangkan konteks penulisan[4].

2.2 Natural Language Processing (NLP) untuk Grammar Checking

NLP merupakan bidang dalam AI yang berfokus pada interaksi antara komputer dan bahasa manusia. NLP digunakan untuk mengurai struktur kalimat, memahami makna, dan memberikan analisis linguistik pada teks[5][6][10]. Dalam konteks grammar checking, NLP memungkinkan sistem mengenali kesalahan seperti subject-verb agreement, tense, penggunaan preposisi, dan sebagainya. Proses ini melibatkan tahap-tahap seperti tokenisasi, parsing, part-of-speech tagging, dan semantic analysis[6][11].

Menurut [12], pendekatan NLP dapat diklasifikasikan ke dalam tiga pendekatan utama, yaitu pendekatan simbolik, statistik, dan koneksionis. Ketiganya memungkinkan komputer memahami dan memproses teks alami dengan lebih dalam. Pendekatan koneksionis, khususnya, memanfaatkan teknik representasi pembelajaran untuk mentransformasikan data linguistik yang kompleks menjadi representasi yang dapat diolah mesin. Hal ini sangat penting dalam sistem grammar checker berbasis AI agar dapat menangani kompleksitas kalimat yang tinggi.

Berbagai penelitian menunjukkan bahwa integrasi NLP dengan machine learning seperti Naïve Bayes, SVM, dan Deep Learning dapat meningkatkan akurasi sistem grammar

checking [7][13]. NLP juga telah banyak diterapkan dalam aplikasi chatbot, sistem penilaian otomatis, dan deteksi plagiarisme[11][14].

2.3 Pengembangan Aplikasi Berbasis Flask

Flask merupakan framework web berbasis Python yang ringan dan fleksibel, banyak digunakan dalam pengembangan aplikasi berbasis AI[5][9]. Flask memungkinkan pengembang untuk membangun RESTful API, mengintegrasikan model AI, serta mengelola frontend dan backend secara modular. Dalam penelitian sebelumnya, Flask digunakan sebagai backend utama dalam sistem anotasi data, klasifikasi teks, hingga aplikasi pendeteksi warna[8][5].

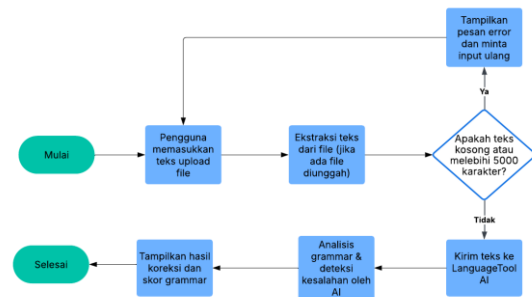
Penggunaan Flask dalam proyek ini dipilih karena sifatnya yang ringan dan mudah dikombinasikan dengan pustaka NLP seperti spaCy dan LanguageTool[9]. Kombinasi Flask dan teknologi AI telah terbukti mampu menciptakan aplikasi web cerdas yang efisien dan skalabel.

2.4 Rekayasa Perangkat Lunak dalam Proyek AI

Penerapan metode rekayasa perangkat lunak, seperti model V-Model dalam pengembangan proyek AI, sangat penting untuk memastikan kualitas sistem[5]. Proses seperti perencanaan kebutuhan, pemodelan sistem, pengujian white-box dan black-box, serta deployment, diterapkan dalam tahapan siklus hidup perangkat lunak[11].

Beberapa penelitian menerapkan pendekatan sistematis dalam pengembangan aplikasi berbasis AI, dimulai dari pengumpulan data, preprocessing, pemodelan AI, hingga penyajian antarmuka pengguna yang interaktif [7][14][15]. Studi juga menunjukkan bahwa keterpaduan antara proses rekayasa perangkat lunak dan teknik AI dapat menghasilkan sistem yang adaptif dan responsif terhadap kebutuhan pengguna.

3. METODE PENELITIAN



Gambar 1. Flowchart Sistem Pemeriksa Tata Bahasa Inggris

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan model Waterfall sebagai dasar dalam proses pengembangan sistem. Model ini dipilih karena memberikan alur kerja yang sistematis dan terstruktur dari tahap awal hingga akhir. Tahapan dalam pendekatan ini meliputi: (1) analisis kebutuhan, (2) perancangan sistem, (3) implementasi, (4) pengujian, dan (5) pemeliharaan. Setiap tahapan dilakukan secara berurutan dan saling berkesinambungan untuk memastikan kualitas aplikasi yang dikembangkan.

3.1. Analisis Kebutuhan

Pengembangan aplikasi ini bertujuan untuk mempermudah pengguna dalam memeriksa kesalahan tata bahasa Inggris secara otomatis melalui antarmuka web yang sederhana. Sistem dirancang agar dapat memproses teks dari input langsung maupun file dokumen (.pdf, .docx, .txt).

Adapun kebutuhan utama sistem meliputi:

1. Pengguna dapat mengetik teks langsung atau mengunggah file dokumen.
2. Sistem menggunakan AI untuk mendeteksi kesalahan tata bahasa dan memberikan saran perbaikan.
3. Hasil koreksi disajikan dalam bentuk daftar kesalahan, saran, dan skor tata bahasa.
4. Aplikasi dapat diakses melalui perangkat desktop maupun mobile.
5. Maksimal 5000 karakter per input dan validasi hanya untuk format file tertentu.

3.2. Perancangan Sistem

Perancangan sistem dilakukan untuk memvisualisasikan alur kerja dan struktur dari aplikasi. Sistem ini dirancang berbasis arsitektur client-server, di mana pengguna mengakses aplikasi melalui browser dan server akan memproses data masukan serta memberikan hasil koreksi. Alur kerja sistem mencakup beberapa tahap utama, yaitu input teks atau file oleh pengguna, pemrosesan dan ekstraksi teks di server, analisis grammar menggunakan modul AI, dan pengembalian hasil berupa umpan balik dan skor. Antarmuka pengguna dirancang menggunakan HTML dan CSS dengan pendekatan desain yang sederhana namun estetik. Komponen utama meliputi: (1) form input teks, (2) tombol unggah file, (3) tombol proses grammar check, dan (4) area hasil koreksi. Struktur folder pada proyek mengikuti standar Flask dengan pemisahan antara file backend (app.py), template HTML (folder templates), dan file statis (folder static).

3.3. Implementasi Sistem

Implementasi sistem dilakukan menggunakan bahasa pemrograman Python dengan framework Flask sebagai backend utama. Integrasi artificial intelligence dilakukan melalui library `language_tool_python` yang berfungsi sebagai pemeriksa tata bahasa berbasis teknologi LanguageTool. File teks yang diunggah diproses menggunakan library `docx2txt` untuk file .docx dan `PyPDF2` untuk file .pdf. Sistem membaca input dari form HTML dan menyimpannya secara sementara untuk dianalisis. Setelah dianalisis, hasil koreksi grammar, termasuk konteks kesalahan, saran perbaikan, dan penilaian skor tata bahasa, akan ditampilkan ke pengguna. Proses ini dilakukan secara otomatis dan real-time setelah pengguna mengirimkan data.

3.4. Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan bahwa sistem bekerja sesuai dengan kebutuhan yang telah ditentukan. Metode pengujian yang digunakan adalah Black-box Testing, yaitu dengan menguji fungsi-fungsi aplikasi tanpa melihat kode program. Pengujian dilakukan dengan skenario sebagai berikut:

1. Memasukkan teks manual dan memastikan sistem mampu memproses

dan memberikan umpan balik yang sesuai.

2. Mengunggah file .pdf, .docx, dan .txt lalu memverifikasi bahwa sistem mampu mengekstraksi dan menganalisis isi file.
3. Menguji input kosong dan input yang melebihi batas karakter untuk memverifikasi validasi sistem.
4. Menguji ketepatan feedback grammar terhadap kalimat yang diketahui memiliki kesalahan tata bahasa.

3.5. Pemeliharaan

Pemeliharaan Sistem dilakukan untuk memastikan aplikasi tetap berjalan dengan optimal setelah tahap implementasi. Kegiatan ini mencakup perbaikan bug yang ditemukan pasca-pengujian, peningkatan performa sistem, serta penyesuaian terhadap masukan pengguna. Selain itu, pemeliharaan juga mencakup pembaruan modul AI LanguageTool jika terdapat versi terbaru yang lebih akurat atau efisien, serta pengelolaan penyimpanan file hasil unggahan agar tidak menumpuk dan membebani server.

4. HASIL DAN PEMBAHASAN

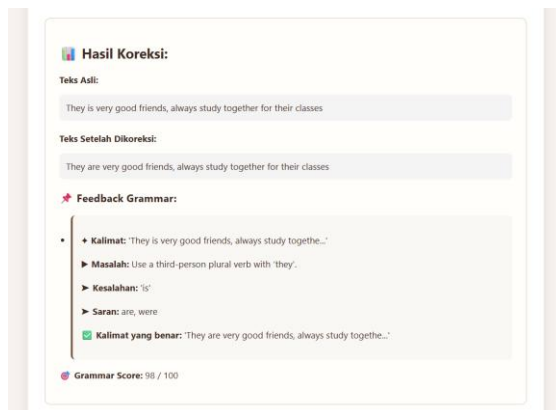
4.1 Hasil Implementasi Sistem

Hasil implementasi sistem menunjukkan bahwa aplikasi *Pemeriksa Grammar Bahasa Inggris* berbasis web berhasil dikembangkan dengan tampilan antarmuka yang sederhana, bersih, dan mudah digunakan. Gambar 2 menunjukkan halaman utama dari aplikasi yang menyediakan dua jenis input: melalui pengetikan langsung pada form teks maupun melalui unggahan file (.txt, .pdf, .docx). Desain ini bertujuan untuk memberikan fleksibilitas kepada pengguna dalam melakukan pemeriksaan grammar sesuai kebutuhan.



Gambar 2. Halaman utama aplikasi (form input teks dan upload file)

Setelah pengguna memasukkan atau mengunggah teks, aplikasi memproses masukan tersebut dan menampilkan hasil pemeriksaan grammar secara detail. Sebagaimana terlihat pada Gambar 3, sistem akan menampilkan teks asli, teks hasil koreksi otomatis, daftar feedback grammar, serta skor grammar yang dihitung berdasarkan jumlah kesalahan yang terdeteksi. Skor ini memberikan indikasi seberapa baik struktur grammar dari teks yang diperiksa.



Gambar 3. Tampilan hasil pemeriksaan grammar

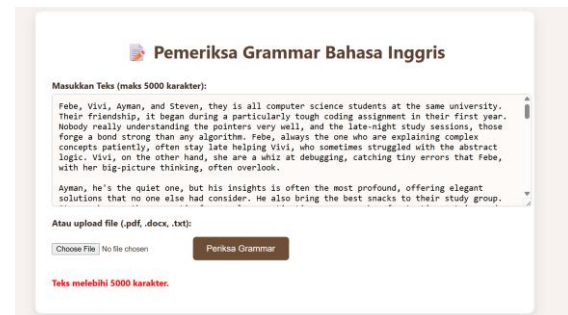
Untuk menjamin keandalan sistem, aplikasi juga menerapkan validasi input yang ketat. Jika pengguna tidak mengisi form atau tidak mengunggah file, maka sistem akan memberikan peringatan berupa pesan error “Teks tidak boleh kosong” seperti ditampilkan pada Gambar 4. Pesan ini ditampilkan dalam warna merah untuk memberikan kejelasan visual dan mencegah pengguna melanjutkan proses tanpa input yang valid.



Gambar 4. Pesan error saat input kosong

Demikian pula, sistem membatasi jumlah karakter maksimum hingga 5000. Jika melebihi batas tersebut, sistem akan memberikan pesan

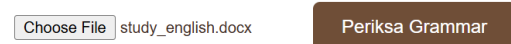
error “Teks melebihi 5000 karakter” (Gambar 5). Hal ini dilakukan untuk menjaga performa pemrosesan dan mencegah beban berlebih pada library pemroses bahasa (LanguageTool).



Gambar 5. Pesan error saat melebihi batas karakter

Proses unggah file berjalan dengan lancar. Gambar 6 menunjukkan bahwa aplikasi mampu menangani file unggahan dan mengekstrak isi dokumen dengan baik, termasuk file PDF, DOCX, dan TXT, lalu menggabungkannya ke dalam input pemeriksaan grammar.

Atau upload file (.pdf, .docx, .txt):



Gambar 6. Proses upload file berjalan dengan baik

4.2 Analisis Fungsionalitas Fitur

Fitur-fitur utama dalam aplikasi ini telah dianalisis berdasarkan pengujian fungsional dan hasil implementasi. Fitur pertama yaitu input teks langsung, bekerja sesuai harapan. Form input memfasilitasi pengguna untuk mengetik atau menempelkan teks Bahasa Inggris secara langsung hingga batas maksimal 5000 karakter. Mekanisme validasi memastikan bahwa hanya teks yang valid yang dapat dikirim untuk diproses. Ini menjamin bahwa hasil koreksi yang ditampilkan adalah representasi dari input yang tepat.

Fitur kedua yaitu unggah file (.pdf, .docx, .txt), berfungsi secara andal. File yang diunggah diproses menggunakan pustaka pemrosesan dokumen seperti docx, PyPDF2, dan Python I/O standar untuk .txt. Setelah file diunggah, sistem akan secara otomatis mengekstrak isi file dan melakukan proses grammar checking menggunakan LanguageTool. Proses ini

berjalan secara transparan di latar belakang dan tidak memerlukan interaksi tambahan dari pengguna.

Fitur ketiga adalah pengecekan grammar dan pemberian feedback detail. Sistem memanfaatkan `language_tool_python` untuk mendeteksi kesalahan grammar dan memberikan saran perbaikan. Hasil feedback tidak hanya menampilkan pesan kesalahan umum, namun juga menyertakan:

- Kalimat asal yang mengandung kesalahan.
- Bagian yang salah (kata/frasa).
- Penjelasan kesalahan.
- Daftar saran perbaikan.
- Kalimat yang sudah dikoreksi secara otomatis.

Fitur keempat adalah skoring grammar. Sistem memberikan skor grammar antara 0 hingga 100 berdasarkan jumlah kesalahan yang ditemukan. Setiap kesalahan mengurangi nilai sebanyak dua poin, sehingga skor menjadi indikator kuantitatif terhadap kualitas tata bahasa teks tersebut.

Selain itu, aplikasi menyediakan tanggapan visual seperti pewarnaan pesan error dan pemisahan hasil (teks asli vs. teks yang telah dikoreksi) yang membantu pengguna memahami dengan cepat bagian-bagian mana yang perlu diperbaiki.

Dengan fungsionalitas yang solid dan tampilan antarmuka yang ramah pengguna, aplikasi ini memenuhi kriteria perangkat lunak edukatif berbasis AI yang dapat digunakan oleh pelajar dan mahasiswa dalam meningkatkan kemampuan grammar Bahasa Inggris mereka.

4.3 Hasil Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode Black Box Testing, dengan tujuan untuk memastikan bahwa setiap fitur pada aplikasi berjalan sesuai dengan fungsinya. Pengujian ini berfokus pada keluaran sistem berdasarkan berbagai masukan pengguna tanpa mempertimbangkan logika internal dari kode program.

Berikut adalah tabel pengujian terhadap fitur-fitur utama:

Tabel 1. Hasil Pengujian Black Box pada Fitur Aplikasi

No.	Fitur yang Diuji	Hasil yang Diharapkan	Hasil
-----	------------------	-----------------------	-------

1.	Input teks langsung	Teks dimasukkan manual dapat diperiksa dan ditampilkan hasil koreksinya	Sukses
2.	Batasan karakter	Teks yang melebihi 5000 karakter ditolak dengan pesan error	Sukses
3.	Input kosong	Input kosong ditolak dengan peringatan "Teks tidak boleh kosong"	Sukses
4.	Upload file .txt	Isi file terbaca dan dikoreksi grammar-nya	Sukses
5.	Upload file .pdf	Isi file terbaca dan hasil grammar ditampilkan	Sukses
6.	Upload file .docx	Isi file Word terbaca dan diproses sistem	Sukses
7.	Upload file tidak didukung	File selain .txt, .pdf, dan .docx ditolak dengan pesan error	Sukses
8.	Penilaian skor grammar	Sistem menghitung skor sesuai jumlah kesalahan	Sukses

Hasil pengujian menunjukkan bahwa seluruh fitur sistem berfungsi dengan baik dan sesuai harapan. Tidak ditemukan error mayor selama pengujian berlangsung. Validasi masukan, pembacaan file, serta pemberian feedback grammar telah berjalan secara optimal. Hal ini memperkuat bahwa sistem dapat dioperasikan dengan baik oleh pengguna umum tanpa memerlukan konfigurasi tambahan.

4.4 Pembahasan Temuan dan Implikasi Pemeliharaan Sistem

Berdasarkan hasil pengembangan dan pengujian sistem, ditemukan bahwa integrasi antara *web framework Flask* dan library *language_tool_python* memberikan hasil yang

cukup optimal dalam melakukan pemeriksaan grammar Bahasa Inggris secara real-time. Aplikasi ini mampu menangani berbagai format input, memberikan umpan balik yang informatif, serta memiliki antarmuka yang responsif dan user-friendly.

Namun, seiring dengan penggunaan yang terus meningkat dan potensi penerapan dalam skala lebih besar (misalnya di lingkungan pendidikan atau lembaga kursus), maka diperlukan pendekatan pemeliharaan (maintenance) yang direncanakan dengan baik. Berdasarkan model rekayasa perangkat lunak waterfall, tahapan pemeliharaan menjadi aspek penting setelah tahap implementasi selesai. Berikut beberapa aspek penting dalam pemeliharaan sistem:

1. Pemeliharaan Korektif

Jika ditemukan bug baru atau error pada sistem—seperti ketidakmampuan membaca file tertentu atau salah deteksi grammar—maka pengembang harus memperbaikinya dengan cepat berdasarkan laporan pengguna.

2. Pemeliharaan Adaptif

Seiring perkembangan teknologi dan browser, sistem perlu diperbarui agar tetap kompatibel. Misalnya, update Flask, library PyPDF2, atau dukungan terhadap format file tambahan seperti .odt.

3. Pemeliharaan Perfektif

Berdasarkan masukan pengguna, sistem dapat ditingkatkan seperti menambahkan fitur highlight kesalahan grammar langsung pada teks, mendukung pengecekan teks multi-bahasa, atau menyimpan riwayat hasil pemeriksaan.

4. Pemeliharaan Preventif

Untuk menjaga keamanan dan performa, perlu dilakukan audit berkala terhadap dependency (pustaka pihak ketiga), serta optimasi kode agar tetap efisien dan terhindar dari celah keamanan.

Dengan mempertimbangkan aspek-aspek tersebut, sistem ini memiliki potensi untuk terus dikembangkan menjadi aplikasi berbasis web yang lebih canggih dan adaptif. Strategi pemeliharaan yang baik akan menjaga keberlangsungan sistem dan meningkatkan kepuasan pengguna dalam jangka panjang.

KESIMPULAN

- Aplikasi *Pemeriksa Grammar Bahasa Inggris* telah berhasil dikembangkan dengan menggunakan framework Flask sebagai basis pengembangan web, serta library `language_tool_python` sebagai komponen pemeriksa tata bahasa berbasis Artificial Intelligence. Aplikasi ini mampu memproses masukan berupa teks langsung maupun file dokumen (.txt, .pdf, dan .docx) secara efektif.
- Fitur utama yang diimplementasikan meliputi pemeriksaan struktur grammar, pemberian umpan balik kesalahan secara rinci, saran perbaikan kalimat, serta perhitungan skor grammar secara otomatis. Seluruh fungsionalitas ini dirancang dengan antarmuka web yang responsif dan mudah dioperasikan oleh pengguna akhir.
- Berdasarkan hasil pengujian dengan metode black box, seluruh fitur sistem berjalan sesuai dengan spesifikasi yang direncanakan. Tidak ditemukan kegagalan pada saat pengujian, baik dalam skenario input teks normal, input kosong, unggahan file valid maupun tidak valid, serta batasan panjang teks.
- Aplikasi ini memiliki potensi untuk dikembangkan lebih lanjut, antara lain dengan menambahkan fitur penyorotan kesalahan langsung pada teks, dukungan bahasa lain selain Bahasa Inggris, penyimpanan riwayat koreksi untuk masing-masing pengguna, serta peningkatan performa pemrosesan teks dalam jumlah besar guna meningkatkan skalabilitas sistem.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, motivasi, dan fasilitas selama proses penyusunan artikel ini. Ucapan terima kasih juga disampaikan kepada seluruh pihak yang terlibat secara langsung maupun tidak langsung dalam pengembangan dan pengujian sistem, sehingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] Sastaparamitha, N. N. A. J., Martarini, N. M. L., Suryawan, I. W. D., Tonyjanto, C., & Datya, A. I. (2022). IMPLEMENTASI APLIKASI

- KAHOOT! TERHADAP HASIL BELAJAR “GRAMMAR” PADA MATA KULIAH BAHASA INGGRIS I DI MAHASISWA ANGKATAN 2020 JURUSAN TEKNIK INFORMATIKA STMIK STIKOM INDONESIA. *Jurnal Teknologi Informasi dan Komputer*, 8(1).
- [2] Sribagus, S. (2018). Kompleksnya Writing Dibandingkan Speaking. *Jurnal Ilmiah Profesi Pendidikan*, 3(2).
- [3] Nurhalimah, I., Iswara, P. D., & Aeni, A. N. (2024). Pengembangan aplikasi android dalam pembelajaran menulis cerita di Sekolah Dasar. *Academy of Education Journal*, 15(1), 102-120.
- [4] Marliyanda, A., Wachyudi, K., & Kartini, D. (2022). Analisis Survei Terhadap Pengguna Grammarly. *Jurnal Educatio Fkip Unma*, 8(3), 1147-1152.
- [5] Supriyanto, R., & Siahaan, D. O. (2024). Sistem Informasi Anotasi Data Penelitian: Pengolahan Data. *Jurnal Teknik ITS*, 13(1), A60-A66.
- [6] Ramadhandi, N. (2023). Penerapan Metode Natural Language Processing (NLP) Pada Question Answering System Untuk Media Informasi Mahasiswa Universitas Bhayangkara Jakarta Raya. *Journal of Informatic and Information Security*, 4(2), 175-186.
- [7] Dermawan, F. D. (2024). Implementasi Artificial Intelligence dalam Pembuatan Website Klasifikasi Genre Buku. *bit-Tech*, 7(2), 618-627.
- [8] Yeza, M. P., Alfrida, M. H., Ramdani, A. L., Saputra, F. A., Susanto, C. P., Giri, E. P., & Mindara, G. P. (2024). Aplikasi Website dengan Flask dan Open CV untuk Filtering Warna Bagi Penderita Buta Warna. *Jurnal Teknik Mesin, Industri, Elektro dan Informatika*, 3(4), 178-189.
- [9] Legi, M., Sengkey, D. F., & Sambul, A. M. (2024). Aplikasi Kecerdasan Artifisial Generatif Untuk Asesmen Pembelajaran Berdasarkan Rubrik: An Application of Generative Artificial Intelligence for Automated Rubric-Based Grading. *Jurnal Teknik Informatika*, 19(3), 183-192.
- [10] Ghirrid, A. A., Sari, R. T. K., & Aldisa, R. T. (2024). Algoritma Natural Language Processing Untuk Aplikasi Penerjemah (Indonesia-Jawa) Menggunakan Metode Speech Processing. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 8(3), 746-759.
- [11] Mubarak, M. I., & Tanjung, M. A. P. (2024). IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PERANCANGAN APLIKASI CHATBOT PADA FIKTI UMSU. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(6), 11992-12001.
- [12] Sakti, K. E., Mardiana, M., & Pradipta, R. A. (2023). RANCANG BANGUN APLIKASI WEB PENDETEKSI WARNA PADA PIXEL GAMBAR DENGAN KNN CLASSIFIER. *Jurnal Informatika dan Teknik Elektro Terapan*, 11(2).
- [13] Zahro, A., Findasari, F., Nursani, I. I., & Bahari, M. F. (2024). MEMBANGUN SITUS WEB CHATBOT UNTUK MENGOPTIMALKAN URBAN FARMING DENGAN INTEGRASI AMAZON BEDROCK. *JURNAL ILMU KOMPUTER DAN MATEMATIKA*, 5(2), 86-96.
- [14] Ananta, Y. E., Yuniati, D., Rolliawati, D., Kunaefi, A., & Permadi, A. (2025). Pengembangan Aplikasi Sido Chatbot sebagai Aplikasi Pengenalan Objek Wisata Kediri Menggunakan Rule-Based Pattern Matching. *Jurnal Teknologi dan Informasi*, 15(1), 40-53.
- [15] Yoanda, Y. P., Nurmallasari, N., & Hidayat, T. (2022). Rancang Bangun Chatbot Untuk Meningkatkan Pelayanan Customer Pada Aplikasi Traveloka. *Jurnal Teknologi Sistem Informasi*, 3(2), 337-352.