

INTEGRASI NATURAL LANGUAGE PROCESSING DAN K-MEANS CLUSTERING PADA SISTEM EVALUASI PEMBELAJARAN BERBASIS REAL-TIME

¹⁾Arya Talaneko, ²⁾Kevin Rinaldy Nizar S*, ³⁾Herry Langi, ⁴⁾Ronny Katuuk, ⁵⁾Harson Kapoh
^{1,2,3,4)}Program Studi D4 Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Manado
¹⁾aryatalaneko96@gmail.com, ²⁾k.vinizar@gmail.com, ³⁾herrylangi@gmail.com,
⁴⁾katuukronny@gmail.com, ⁵⁾hvskapoh@gmail.com
*Corresponding Author: k.vinizar@gmail.com

INFO ARTIKEL

Riwayat Artikel :

Diterima : 19 Juni 2026

Disetujui : 5 Agustus 2026

Kata Kunci :

NLP, K-Means Clustering,
evaluasi pembelajaran, TF-IDF,
Cosine Similarity.

ABSTRAK

Penelitian ini bertujuan mengembangkan sistem evaluasi pembelajaran berbasis web yang mengintegrasikan Natural Language Processing dan K-Means Clustering secara real-time. Permasalahan utama yang diangkat adalah koreksi esai manual yang membutuhkan waktu lama, potensi subjektivitas penilaian, serta belum tersedianya sistem yang mampu memetakan performa akademik siswa secara otomatis. Sistem dikembangkan menggunakan arsitektur hibrida dengan Laravel sebagai aplikasi utama dan Python Flask sebagai layanan komputasi kecerdasan buatan. Penilaian esai dilakukan menggunakan TF-IDF dan Cosine Similarity untuk menghitung kemiripan jawaban siswa dengan kunci jawaban guru, sedangkan K-Means Clustering digunakan untuk mengelompokkan siswa berdasarkan nilai akhir, nilai pilihan ganda, dan nilai esai. Hasil implementasi menunjukkan bahwa sistem mampu menampilkan nilai ujian, status kelulusan, dan kategori performa siswa secara real-time. Sistem ini dapat membantu guru mempercepat proses evaluasi dan mendukung pengambilan keputusan pembelajaran berbasis data.

ARTICLE INFO

Article History :

Received : Jun 19, 2026

Accepted : Aug 5, 2026

Keywords:

NLP, K-Means Clustering,
learning evaluation, TF-IDF,
Cosine Similarity.

ABSTRACT

This study aims to develop a web-based learning evaluation system that integrates Natural Language Processing and K-Means Clustering in real time. The main problems addressed in this study are time-consuming manual essay grading, potential subjectivity in assessment, and the absence of a system capable of automatically mapping students' academic performance. The system was developed using a hybrid architecture, with Laravel as the main application and Python Flask as the artificial intelligence computation service. Essay assessment was carried out using TF-IDF and Cosine Similarity to measure the similarity between students' answers and the teacher's answer key, while K-Means Clustering was used to group students based on final scores, multiple-choice scores, and essay scores. The implementation results show that the system can display exam scores, passing status, and student performance categories in real time. This system can help teachers accelerate the evaluation process and support data-driven learning decisions.

1. PENDAHULUAN

Perkembangan teknologi digital mendorong perubahan dalam pelaksanaan evaluasi pembelajaran di sekolah. Evaluasi yang sebelumnya dilakukan secara manual mulai dialihkan ke sistem berbasis web karena prosesnya lebih cepat, terdokumentasi, dan mudah diakses. Sistem evaluasi digital tidak hanya digunakan sebagai media pengerjaan soal. Sistem juga dapat mengelola bank soal, menyimpan jawaban, menghitung nilai, menampilkan riwayat hasil belajar, dan menyediakan informasi akademik bagi guru. Model asesmen digital dapat meningkatkan efisiensi serta mempercepat penyampaian hasil evaluasi kepada peserta didik (Susilawati et al., 2025).

Salah satu persoalan dalam evaluasi pembelajaran adalah pemeriksaan jawaban esai. Soal esai dapat mengukur kemampuan siswa dalam memahami konsep, menjelaskan hubungan antargagasan, dan menyusun jawaban secara mandiri. Pemeriksaan secara manual membutuhkan waktu yang cukup panjang ketika jumlah peserta dan butir soal meningkat. Skor yang diberikan juga dapat berbeda karena dipengaruhi oleh kelelahan pemeriksa, perbedaan interpretasi, serta konsistensi rubrik penilaian. Kondisi tersebut mendorong pengembangan Automated Essay Scoring atau AES sebagai alat bantu penilaian berbasis Natural Language Processing.

Penelitian terdahulu pada kelompok pertama menggunakan pendekatan kemiripan leksikal berbasis TF-IDF dan Cosine Similarity. Mi'andri, Siregar and Utami (2021) mengembangkan sistem penilaian otomatis untuk jawaban esai mahasiswa menggunakan TF-IDF dan Vector Space Model. Jawaban mahasiswa direpresentasikan sebagai vektor dan dibandingkan dengan kunci jawaban untuk memperoleh tingkat kemiripan. Penelitian tersebut memperlihatkan bahwa pendekatan berbasis vektor dapat membantu mempercepat pemeriksaan jawaban. Keterbatasannya terletak pada ketergantungan terhadap kesamaan kata. Jawaban yang benar secara konsep dapat memperoleh kemiripan rendah ketika menggunakan susunan kata atau istilah yang berbeda.

Permata, Rendika and Julianty (2025) mengembangkan sistem evaluasi esai dengan memanfaatkan Bag of Words, TF-IDF, Cosine Similarity, dan Manhattan Distance. Objek yang dinilai berupa jawaban esai yang dibandingkan dengan jawaban acuan. Hasil penelitian menunjukkan bahwa Cosine Similarity memberikan kesesuaian yang lebih baik dibandingkan Manhattan Distance dalam mengukur kedekatan jawaban. Metode tersebut memiliki proses komputasi yang relatif ringan dan mudah diterapkan pada aplikasi berbasis web. Keterbatasannya masih berkaitan dengan representasi kata yang belum sepenuhnya memahami sinonim, parafrasa, urutan kata, dan konteks kalimat.

Susilawati *et al.* (2025) mengembangkan model asesmen digital untuk penilaian esai pendek secara otomatis pada program studi pendidikan agama Islam. Sistem tersebut diarahkan untuk mempercepat proses penilaian dan meningkatkan konsistensi pemeriksaan. Hasilnya memperlihatkan bahwa asesmen digital dapat mendukung efisiensi kerja pengajar. Pengujian pada mata pelajaran, karakteristik jawaban, dan institusi yang berbeda masih dibutuhkan untuk mengetahui tingkat generalisasi model. Temuan dari kelompok penelitian ini menunjukkan bahwa TF-IDF dan Cosine Similarity sesuai digunakan sebagai metode dasar yang ringan. Metode tersebut belum dapat menggantikan kemampuan penilai dalam memahami seluruh makna jawaban.

Kelompok penelitian kedua mengembangkan AES berbasis semantic similarity dan Transformer. Pradani and Suadaa (2023) menggunakan Semantic Textual Similarity berbasis Transformer untuk menilai jawaban esai bahasa Indonesia. Model Transformer dibandingkan dengan pendekatan TF-IDF dan Cosine Similarity sebagai baseline. Hasil penelitian memperlihatkan bahwa representasi kontekstual dapat membaca hubungan makna yang tidak selalu ditunjukkan oleh kesamaan kata. Model semantik membutuhkan data pelatihan, sumber daya komputasi, dan proses implementasi yang lebih besar dibandingkan metode berbasis TF-IDF.

Amalia et al. (2024) mengembangkan penilaian dan umpan balik esai bahasa Indonesia menggunakan Bidirectional Encoder

Representations from Transformers. Model IndoBERT digunakan untuk mempelajari representasi kontekstual jawaban dan menghasilkan skor yang mendekati penilaian manusia. Penelitian tersebut menunjukkan bahwa Transformer berpotensi meningkatkan kesesuaian antara skor otomatis dan skor penilai. Ruang lingkup data yang terbatas pada jenis esai dan konteks tertentu masih membatasi penerapan model pada mata pelajaran lain.

Aliyah et al. (2025) menggabungkan IndoBERT dan IndoSBERT melalui pendekatan transfer learning untuk penilaian esai bahasa Indonesia. IndoBERT digunakan untuk memperoleh representasi kontekstual, sedangkan IndoSBERT digunakan untuk memperkuat representasi kemiripan pada tingkat kalimat. Kombinasi tersebut mampu menghasilkan representasi semantik yang lebih kaya daripada metode berbasis frekuensi kata. Penerapannya memerlukan data berlabel, proses fine-tuning, perangkat komputasi yang lebih tinggi, dan pengaturan model yang lebih kompleks.

Prastyo et al. (2025) membandingkan beberapa pretrained Transformer untuk Automated Essay Scoring bahasa Indonesia. Hasil penelitian menunjukkan bahwa pemilihan model pralatih memengaruhi performa penilaian. Tidak semua model memberikan hasil yang sama pada data dan karakteristik jawaban yang berbeda. Penelitian tersebut juga menegaskan bahwa performa Transformer bergantung pada kualitas data, distribusi skor, panjang teks, dan kesesuaian domain pelatihan.

Sufandi et al. (2026) membandingkan beberapa model Transformer pada 3.077 esai dengan dua jawaban referensi. Hasil perbandingan menunjukkan adanya variasi kemampuan model dalam mengukur kemiripan dan memberikan skor. Model Transformer lebih baik dalam menangkap konteks dibandingkan metode berbasis kata. Hasilnya tetap dipengaruhi oleh karakteristik jawaban referensi, jumlah data, rubrik, dan domain pertanyaan. Pemanfaatan model tersebut juga perlu mempertimbangkan transparansi keputusan dan kemampuan guru dalam meninjau kembali skor yang dihasilkan sistem.

Kajian Li & Ng (2024) menjelaskan bahwa perkembangan AES telah bergerak dari

penggunaan fitur leksikal menuju neural network, Transformer, dan large language model. Persoalan validitas, interpretabilitas, generalisasi lintas pertanyaan, dan kesesuaian skor dengan penilai manusia masih menjadi tantangan. Misgna et al. (2025) juga menemukan bahwa model deep learning mampu mempelajari pola kompleks dalam esai, tetapi alasan yang digunakan model dalam menghasilkan skor sering sulit dijelaskan. Faseeh et al. (2024) menawarkan pendekatan hibrida dengan menggabungkan RoBERTa embedding, fitur linguistik, dan Lightweight XGBoost. Pendekatan tersebut menghasilkan performa tinggi pada data pengujian, tetapi membutuhkan proses komputasi dan rekayasa fitur yang lebih kompleks.

Hasil kajian tersebut memperlihatkan dua pilihan utama dalam pembangunan AES. Metode Transformer memiliki kemampuan semantik yang lebih baik, tetapi membutuhkan sumber daya komputasi dan data pelatihan yang memadai. TF-IDF dan Cosine Similarity lebih ringan, mudah dijelaskan, dan sesuai untuk membangun baseline pada sistem sekolah. Metode berbasis kemiripan kata perlu ditempatkan sebagai alat bantu guru. Hasil penilaian tetap perlu ditinjau ketika jawaban menggunakan parafrasa, sinonim, atau penjelasan yang benar dengan struktur yang berbeda.

Permasalahan evaluasi pembelajaran tidak berhenti pada koreksi jawaban. Guru juga perlu mengetahui pola performa akademik setelah nilai diperoleh. Nilai yang ditampilkan dalam bentuk angka belum selalu memberikan informasi tentang siswa yang memerlukan remedial, pendampingan, atau pengayaan. Educational data mining dapat digunakan untuk mengubah data nilai menjadi kelompok yang lebih mudah ditafsirkan. Salah satu algoritma yang banyak digunakan untuk kebutuhan tersebut adalah K-Means Clustering.

Dacwanda & Nataliani (2021) menerapkan K-Means untuk mengelompokkan siswa berdasarkan nilai pengetahuan dan keterampilan. Hasil penelitian menunjukkan bahwa algoritma tersebut mampu membentuk kelompok siswa dengan karakteristik nilai yang berbeda. Hasil klaster dapat membantu sekolah mengenali pola pencapaian siswa. Penelitian tersebut belum

mengintegrasikan hasil klasterisasi dengan proses ujian dan koreksi esai dalam satu aplikasi.

Hendrastuty (2024) menerapkan K-Means pada data hasil pembelajaran siswa untuk mengidentifikasi pola yang tersembunyi dan membagi siswa berdasarkan tingkat pencapaian. Hasil pengelompokan dapat digunakan sebagai informasi bagi guru dalam mengevaluasi hasil belajar. Penetapan jumlah klaster dan interpretasi setiap klaster masih memerlukan pertimbangan pedagogis agar kelompok yang terbentuk tidak hanya didasarkan pada perhitungan matematis.

Azzahra & Sriani (2025) menggunakan K-Means untuk mengelompokkan nilai akademik siswa sekolah menengah. Data nilai mata pelajaran dikelompokkan berdasarkan kedekatan karakteristiknya. Hasil penelitian memberikan gambaran distribusi kemampuan akademik siswa. Keterbatasannya terletak pada sensitivitas K-Means terhadap pemilihan jumlah klaster, centroid awal, skala data, dan jumlah observasi. Validasi klaster diperlukan agar kelompok yang terbentuk tidak hanya mengikuti konfigurasi awal algoritma.

Penelitian Maghfiroh & Fatah (2025) juga menghasilkan tiga kelompok prestasi siswa, yaitu tinggi, sedang, dan rendah. Pengelompokan tersebut membantu sekolah mengidentifikasi kondisi akademik secara lebih objektif. Fitur yang digunakan masih berpusat pada data nilai. Aspek kehadiran, keterlibatan belajar, pola pengerjaan tugas, serta faktor lain yang memengaruhi pencapaian belum dimasukkan ke dalam proses klasterisasi.

Kelompok penelitian berikutnya berfokus pada learning analytics dan identifikasi dini performa siswa. Santos & Henriques (2023) menggunakan log Learning Management System untuk membangun prediksi performa yang dapat diterapkan pada berbagai mata kuliah. Hasil penelitian menunjukkan bahwa data aktivitas digital dapat digunakan untuk mengenali risiko akademik lebih awal. Model tersebut lebih berorientasi pada prediksi dan belum menggabungkan proses koreksi esai otomatis.

Fazil et al. (2024) mengembangkan model deep learning untuk memprediksi performa siswa menggunakan data registrasi akademik, aktivitas pada virtual learning environment, dan

penilaian tengah semester. Hasilnya memperlihatkan bahwa data keterlibatan dapat membantu membentuk representasi perilaku siswa. Model yang digunakan memiliki tingkat kompleksitas cukup tinggi dan memerlukan data aktivitas yang lengkap.

Almalawi et al. (2024) meninjau 124 penelitian mengenai model prediksi pendidikan. Kajian tersebut menunjukkan bahwa machine learning dapat membantu memprediksi performa, mengenali siswa berisiko, dan mendukung personalisasi pembelajaran. Tantangan yang ditemukan mencakup kualitas data, bias, transparansi, interpretasi model, dan perbedaan proses prapengolahan data.

Dai et al. (2025) mengembangkan learning analytics untuk identifikasi dini siswa berisiko dan pemberian intervensi umpan balik. Hasil penelitian menegaskan bahwa keluaran analitik perlu dihubungkan dengan tindakan pembelajaran agar tidak berhenti pada visualisasi data. Ngulube & Ncube (2025) melalui tinjauan terhadap 41 penelitian menemukan bahwa integrasi learning analytics pada LMS berpotensi meningkatkan pengalaman pengguna dan pengambilan keputusan. Hambatan yang masih ditemukan mencakup generalisasi, interpretasi hasil, kualitas sistem, etika, dan penerapan hasil analitik pada konteks institusi yang berbeda.

Penelitian terdahulu memperlihatkan bahwa koreksi esai otomatis, klasterisasi prestasi siswa, dan learning analytics umumnya dikembangkan sebagai kajian yang terpisah. Penelitian AES berfokus pada kesesuaian skor jawaban dengan kunci atau penilai manusia. Penelitian K-Means lebih banyak menggunakan data nilai yang telah tersedia untuk analisis secara retrospektif. Penelitian learning analytics berfokus pada prediksi risiko dari aktivitas LMS. Belum banyak penelitian yang menggabungkan proses ujian daring, koreksi esai otomatis, perhitungan nilai, penentuan kelulusan, dan pengelompokan performa siswa dalam satu alur aplikasi berbasis web.

Penelitian ini mengisi kesenjangan tersebut melalui integrasi NLP dan K-Means Clustering pada sistem evaluasi pembelajaran berbasis web. TF-IDF dan Cosine Similarity digunakan untuk menilai kemiripan jawaban esai dengan kunci jawaban. K-Means digunakan untuk

mengelompokkan siswa berdasarkan nilai akhir, nilai pilihan ganda, dan nilai esai. Sistem mengubah hasil klaster menjadi kategori Aman, Bimbingan, dan Risiko Tinggi agar lebih mudah digunakan oleh guru.

Kebaruan penelitian terletak pada integrasi dua fungsi analitik dalam satu sistem, yaitu koreksi esai otomatis dan pemetaan performa akademik. Kedua proses dijalankan melalui arsitektur hibrida Laravel dan Python Flask pada studi kasus SMP Katolik St. Johanis Laikit. Istilah real-time dalam penelitian ini merujuk pada kemampuan sistem memproses dan menampilkan nilai, status kelulusan, serta kategori performa segera setelah data ujian dikirim dan algoritma selesai dijalankan. Penelitian ini belum melakukan pengukuran khusus terhadap latency, throughput, atau response time sistem.

Tujuan penelitian adalah mengembangkan sistem evaluasi pembelajaran berbasis web yang mampu memproses jawaban esai, menggabungkan skor esai dan pilihan ganda, serta mengelompokkan performa akademik siswa. Sistem diharapkan dapat mempercepat pengolahan hasil ujian dan memberikan informasi awal bagi guru dalam menentukan pengayaan, pendampingan, atau remedial.

2. METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan bentuk pengembangan dan pengujian proof of concept. Objek penelitian mencakup aplikasi evaluasi pembelajaran berbasis web dan data hasil ujian yang dikelola pada SMP Katolik St. Johanis Laikit. Pendekatan rekayasa perangkat lunak dipilih karena penelitian tidak hanya menganalisis algoritma, tetapi juga merancang arsitektur, basis data, antarmuka, layanan komputasi, dan alur penggunaan sistem. Tahapan penelitian meliputi identifikasi masalah, studi literatur, analisis kebutuhan pengguna, perancangan sistem, implementasi, pengujian fungsional, dan evaluasi hasil implementasi (Maryani et al., 2022).

Kebutuhan sistem diperoleh dari proses evaluasi pembelajaran yang melibatkan admin, guru, dan siswa. Admin membutuhkan fitur pengelolaan pengguna, kelas, mata pelajaran, dan data master. Guru membutuhkan fitur bank

soal, jadwal ujian, koreksi otomatis, rekap nilai, serta analisis performa siswa. Siswa membutuhkan fitur pengerjaan ujian, countdown timer, pengiriman jawaban, dan tampilan hasil. Kebutuhan tersebut digunakan sebagai dasar perancangan use case, basis data, class diagram, activity diagram, dan antarmuka sistem.

Sistem dibangun menggunakan arsitektur hibrida yang memisahkan aplikasi utama dan layanan komputasi algoritma. Laravel 11 digunakan sebagai aplikasi utama untuk autentikasi, pengelolaan pengguna, data kelas, mata pelajaran, soal, jadwal, jawaban, nilai, serta laporan. MySQL digunakan untuk menyimpan data utama dan data transaksi ujian. Python Flask digunakan sebagai layanan terpisah untuk menjalankan koreksi esai dan K-Means Clustering.

Pemisahan layanan dipilih agar pengelolaan transaksi aplikasi dan proses komputasi algoritma dapat dikembangkan secara modular. Laravel berfungsi sebagai client yang mengirimkan data melalui HTTP Request. Flask menerima data dalam format JSON, menjalankan proses algoritma, dan mengembalikan hasil dalam format JSON. Arsitektur microservice dapat mendukung pemisahan fungsi aplikasi, walaupun kompleksitas komunikasi, monitoring, dan pengujian antarlayanan tetap perlu diperhatikan (Dahri et al., 2021).

Antarmuka sistem dikembangkan menggunakan HTML5, Vanilla JavaScript, dan Tailwind CSS. JavaScript digunakan untuk mengatur countdown timer, validasi waktu ujian, penguncian lembar jawaban, dan auto-submit. Tailwind CSS digunakan untuk membangun tampilan responsif bagi admin, guru, dan siswa. Komputasi algoritma menggunakan scikit-learn dan NumPy. Scikit-learn dipertahankan sebagai rujukan teknis karena menyediakan implementasi TF-IDF, Cosine Similarity, StandardScaler, dan K-Means yang digunakan dalam penelitian (Pedregosa et al., 2011).

Penilaian esai otomatis diawali dengan pengiriman kunci jawaban guru dan jawaban siswa dari Laravel ke Flask. Teks diproses melalui case folding, cleaning, tokenizing, dan stopword removal. Case folding mengubah seluruh karakter menjadi huruf kecil. Cleaning menghapus tanda baca, angka, simbol, dan

karakter yang tidak diperlukan. Tokenizing memisahkan teks menjadi unit kata. Stopword removal menghapus kata umum yang memiliki kontribusi informasi rendah.

Tahap prapengolahan digunakan untuk mengurangi variasi teks yang tidak berkaitan langsung dengan isi jawaban. Proses tersebut juga memiliki konsekuensi karena beberapa kata fungsi dapat memiliki peran dalam konteks kalimat. Daftar stopwords perlu disesuaikan dengan karakteristik bahasa Indonesia dan mata pelajaran agar kata yang berkaitan dengan makna jawaban tidak ikut dihapus.

Teks yang telah diproses direpresentasikan menggunakan Term Frequency-Inverse Document Frequency. Term Frequency menunjukkan frekuensi kemunculan sebuah kata dalam dokumen. Inverse Document Frequency menunjukkan tingkat kekhususan kata berdasarkan jumlah dokumen yang memuat kata tersebut. Rumus IDF yang digunakan adalah:

$$IDF(t) = \ln \left(\frac{N}{DF(t)+1} \right) + 1 \quad (1)$$

Keterangan:

$IDF(t)$ adalah nilai inverse document frequency dari kata t , N adalah jumlah seluruh dokumen, sedangkan $DF(t)$ adalah jumlah dokumen yang mengandung kata (t) . Penambahan nilai 1 digunakan untuk menghindari pembagian dengan nol. Vektor TF-IDF kunci jawaban dan jawaban siswa dibandingkan menggunakan Cosine Similarity. Perhitungan dilakukan dengan rumus:

$$Similarity(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \quad (2)$$

Keterangan:

A adalah vektor kunci jawaban, B adalah vektor jawaban siswa, $A \cdot B$ adalah hasil perkalian dot product antara dua vektor, sedangkan $\|A\|$ dan $\|B\|$ merupakan panjang masing-masing vektor. Nilai kemiripan berada pada rentang 0 sampai 1. Nilai yang mendekati 1 menunjukkan kemiripan kata yang lebih tinggi.

TF-IDF dan Cosine Similarity dipilih karena memiliki proses komputasi yang ringan, mudah diimplementasikan, dan dapat dijelaskan kepada pengguna. Metode tersebut banyak digunakan sebagai baseline dalam penilaian teks dan AES

bahasa Indonesia (Permata et al., 2025; Pradani & Suadaa, 2023). Metode ini tidak sepenuhnya memahami sinonim, parafrasa, negasi, hubungan antarkalimat, dan konteks semantik. Skor yang dihasilkan ditempatkan sebagai alat bantu, bukan sebagai pengganti mutlak keputusan guru.

Nilai Cosine Similarity dikonversi menjadi skor berdasarkan bobot soal yang ditentukan oleh guru. Skor setiap jawaban esai dijumlahkan untuk memperoleh nilai esai. Nilai esai kemudian digabungkan dengan nilai pilihan ganda untuk menghasilkan nilai akhir. Sistem membandingkan nilai akhir dengan Kriteria Ketuntasan Minimal untuk menentukan status lulus atau belum lulus.

Pengelompokan performa siswa dilakukan menggunakan K-Means Clustering. Fitur yang digunakan terdiri atas nilai akhir, nilai pilihan ganda, dan nilai esai. Setiap siswa direpresentasikan dalam vektor:

$$X = [Nilai\ Akhir, Nilai\ PG, Nilai\ Esai] \quad (3)$$

Keterangan:

(NA_i) adalah nilai akhir siswa ke-(i), (PG_i) adalah nilai pilihan ganda, dan (ES_i) adalah nilai esai.

Data dinormalisasi menggunakan StandardScaler sebelum proses klasterisasi. Normalisasi dibutuhkan karena perbedaan distribusi dan rentang antarfitur dapat memengaruhi jarak Euclidean. Rumus normalisasi Z-score adalah:

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

Keterangan:

z adalah nilai hasil normalisasi, x adalah nilai asli, μ adalah rata-rata fitur, dan σ adalah standar deviasi.

K-Means dijalankan dengan jumlah klaster ($k=3$). Penetapan tiga klaster mengikuti kebutuhan sistem untuk menghasilkan kategori Aman, Bimbingan, dan Risiko Tinggi. Jarak data dengan centroid dihitung menggunakan Euclidean Distance:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^D (x_{ik} - c_{jk})^2} \quad (5)$$

Keterangan:

$d(x_i, c_j)$ adalah jarak antara data siswa ke- i dan centroid kluster ke- j , D adalah jumlah fitur, x_{ik} adalah nilai fitur ke- k dari siswa ke- i , dan c_{jk} adalah nilai fitur ke- k dari centroid kluster ke- j .

Setiap data ditempatkan pada centroid terdekat. Centroid diperbarui menggunakan rata-rata anggota kluster. Proses tersebut diulang sampai keanggotaan kluster stabil atau perubahan centroid telah memenuhi kriteria berhenti algoritma.

Label kluster ditentukan setelah seluruh proses iterasi selesai. Sistem menghitung rata-rata nilai akhir pada masing-masing kluster. Kluster dengan rata-rata tertinggi diberi label Aman. Kluster dengan rata-rata menengah diberi label Bimbingan. Kluster dengan rata-rata terendah diberi label Risiko Tinggi. Pemberian label dilakukan setelah kluster terbentuk agar nomor kluster yang dihasilkan algoritma tidak langsung dianggap memiliki urutan akademik tertentu.

K-Means dipilih karena mudah diterapkan, efisien pada data numerik, dan banyak digunakan dalam pemetaan performa akademik (Dacwanda & Nataliani, 2021; Azzahra & Sriani, 2025; Hendrastuty, 2024). Hasil K-Means dipengaruhi oleh jumlah data, fitur, skala, centroid awal, dan jumlah kluster. Kategori yang dihasilkan perlu dibaca bersama data akademik lain dan pertimbangan guru.

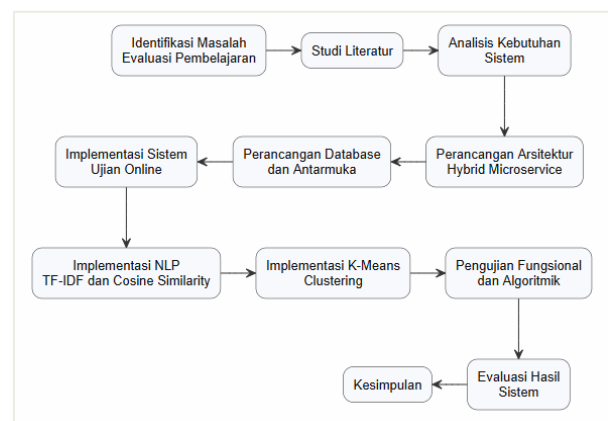
Istilah *real-time* dalam penelitian ini didefinisikan secara operasional sebagai proses ketika nilai, status kelulusan, dan hasil klusterisasi dapat ditampilkan setelah pengguna mengirim jawaban atau menjalankan analisis tanpa rekapitulasi manual. Penelitian belum mengukur latency, response time, throughput, penggunaan memori, atau kemampuan sistem ketika diakses secara bersamaan oleh banyak pengguna.

Pengujian sistem dilakukan menggunakan pengujian fungsional dengan pendekatan black box testing dan pemeriksaan keluaran algoritma. Black box testing digunakan untuk memeriksa kesesuaian fungsi sistem berdasarkan masukan dan keluaran yang dihasilkan tanpa menelaah struktur kode program secara internal. Pendekatan ini sesuai untuk aplikasi berbasis web karena pengujian difokuskan pada respons

setiap fitur terhadap skenario penggunaan yang telah ditentukan (Kaphor et al., 2021). Pengujian fungsional mencakup login, pengelolaan data master, bank soal, jadwal ujian, pengerjaan soal, countdown timer, auto-submit, penyimpanan jawaban, koreksi esai, perhitungan nilai, klusterisasi, penyimpanan hasil, dan ekspor laporan.

Pemeriksaan keluaran algoritma dilakukan dengan mengamati nilai kemiripan esai, skor esai, nilai akhir, dan label kluster yang dihasilkan sistem. Pemeriksaan tersebut bertujuan memastikan bahwa data dapat diproses dari Laravel ke Flask dan dikembalikan ke sistem utama dengan format yang benar.

Ruang lingkup evaluasi penelitian masih berfokus pada keberhasilan integrasi dan fungsi sistem. Penelitian belum membandingkan skor otomatis dengan skor sejumlah guru menggunakan Mean Absolute Error, Root Mean Squared Error, Pearson Correlation, atau Quadratic Weighted Kappa. Penelitian juga belum mengukur kualitas kluster menggunakan Silhouette Score, Davies-Bouldin Index, atau Calinski-Harabasz Index. Keterbatasan tersebut menjadi batas interpretasi terhadap hasil penelitian.



Gambar 1. Alur Penelitian

Alur penelitian dalam pengembangan sistem ini dilakukan melalui sepuluh tahap, yaitu identifikasi masalah, studi literatur, analisis kebutuhan pengguna, perancangan arsitektur sistem, perancangan basis data dan antarmuka, implementasi ujian *online*, implementasi modul *NLP*, implementasi *K-Means Clustering*, pengujian sistem, serta evaluasi hasil.

Penelitian diawali dengan mengkaji masalah koreksi esai manual, keterlambatan

pengolahan nilai, subjektivitas penilaian, dan belum adanya deteksi dini performa akademik siswa. Sistem kemudian dirancang dengan *Laravel* sebagai aplikasi utama dan *Flask* sebagai layanan komputasi kecerdasan buatan. Implementasi mencakup fitur ujian *online*, penilaian esai otomatis menggunakan *preprocessing*, *TF-IDF*, dan *Cosine Similarity*, serta pengelompokan siswa menggunakan *K-Means* berdasarkan nilai akhir, nilai pilihan ganda, dan nilai esai.

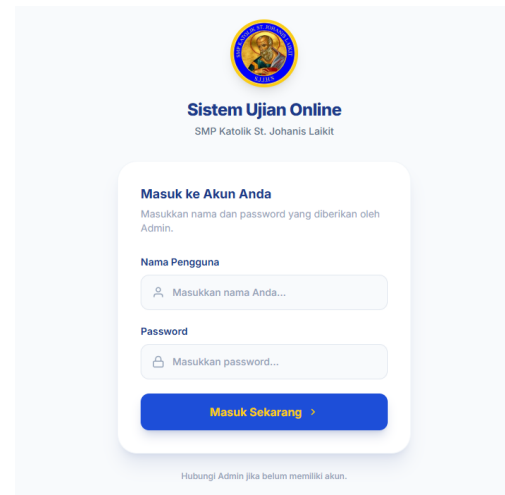
Tahap akhir dilakukan melalui pengujian fungsional dan algoritmik untuk menilai kemampuan sistem dalam mempercepat penilaian dan menampilkan kategori performa akademik siswa secara *real-time*.

3. HASIL DAN PEMBAHASAN

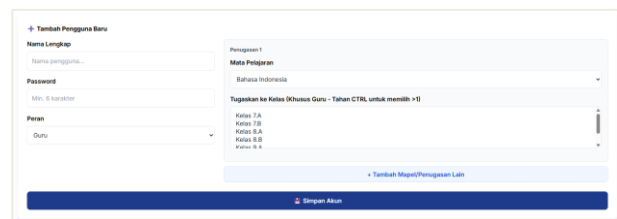
3.1. Hasil Implementasi dan Integrasi Sistem

Hasil penelitian ini berupa sistem evaluasi pembelajaran berbasis web yang terintegrasi dengan teknologi *Natural Language Processing* dan *K-Means Clustering*. Sistem dikembangkan menggunakan arsitektur *hibrida* yang memisahkan aplikasi utama berbasis *Laravel 11* dan layanan komputasi kecerdasan buatan berbasis *Python Flask*. *Laravel* berfungsi sebagai pusat pengelolaan data ujian, autentikasi pengguna, pengaturan jadwal, bank soal, penyimpanan jawaban, rekap nilai, dan tampilan dashboard. *Python Flask* digunakan sebagai *microservice* untuk memproses koreksi esai otomatis dan pengelompokan performa akademik siswa. Integrasi kedua layanan dilakukan melalui pertukaran data berbasis *HTTP Request* dengan format *JSON*.

Secara umum, sistem berhasil menghubungkan tiga komponen utama, yaitu antarmuka pengguna, database *MySQL*, dan layanan kecerdasan buatan. Antarmuka pengguna digunakan oleh admin, guru, dan siswa sesuai hak akses masing-masing. Database *MySQL* digunakan untuk menyimpan seluruh data utama dan data transaksi ujian. Layanan *Flask* menerima data dari *Laravel*, menjalankan proses algoritmik, lalu mengembalikan hasil pemrosesan ke *Laravel* untuk disimpan dan ditampilkan pada dashboard guru. Pola integrasi ini membuat sistem evaluasi tidak hanya berfungsi sebagai aplikasi ujian online, tetapi juga sebagai sistem analitik pembelajaran.

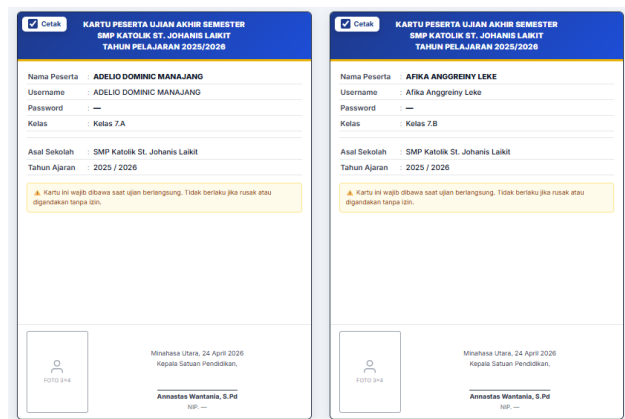


Gambar 2. Tampilan Halaman Login Sistem



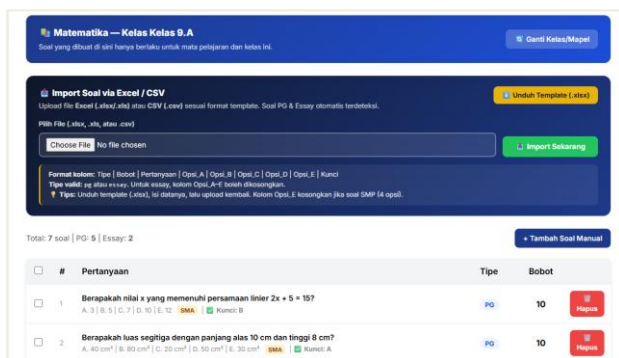
Gambar 3. Tampilan Pengelolaan Data Master

Pada sisi admin, sistem berhasil mengimplementasikan fitur pengelolaan data master, yaitu data kelas, data pengguna, data guru, data siswa, dan mata pelajaran. Admin juga dapat melakukan pemetaan guru ke kelas melalui tabel *teacher_classes*. Fitur ini penting karena sistem menerapkan pembatasan akses data, sehingga guru hanya dapat melihat kelas dan siswa yang menjadi tanggung jawabnya. Admin juga dapat mencetak kartu peserta ujian yang berisi identitas siswa.



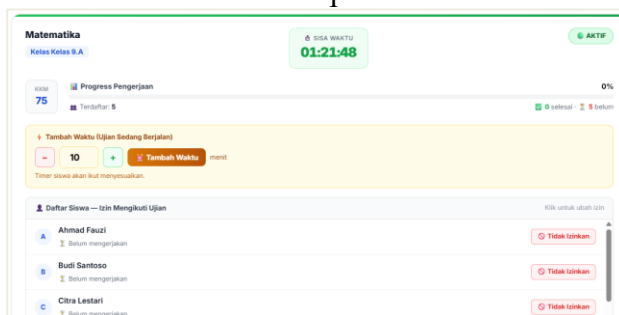
Gambar 4. Kartu Peserta Ujian

Implementasi ini menunjukkan bahwa sistem mendukung administrasi ujian secara lebih tertib dan terdokumentasi.



Gambar 5. Tampilan Halaman Bank Soal Guru

Pada sisi guru, sistem berhasil mengimplementasikan fitur bank soal dinamis. Guru dapat membuat soal pilihan ganda dan soal esai dalam satu paket ujian. Setiap soal dapat diberi bobot nilai berbeda sesuai tingkat kesulitan atau kebutuhan penilaian.



Gambar 6. Tampilan Jadwal Ujian oleh Guru

Guru juga dapat membuat jadwal ujian dengan menentukan mata pelajaran, kelas, durasi, waktu mulai, waktu berakhir, status aktif ujian, dan Kriteria Ketuntasan Minimal. Setelah ujian selesai, guru dapat melihat rekap nilai siswa yang mencakup nilai pilihan ganda, nilai esai, nilai akhir, status kelulusan, serta hasil pengelompokan performa akademik.

Pada sisi siswa, sistem berhasil mengimplementasikan fitur ujian online yang dapat diakses sesuai jadwal yang telah ditentukan guru. Tombol mulai ujian hanya dapat digunakan ketika jadwal ujian aktif dan berada dalam rentang waktu yang sesuai. Saat ujian berlangsung, sistem menampilkan *countdown timer* secara real-time. Ketika waktu habis, sistem melakukan *auto-submit* dan mengunci form jawaban.



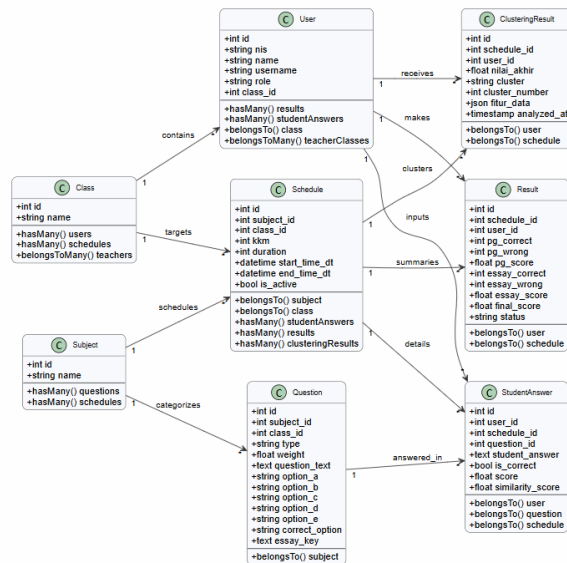
Gambar 7. Tampilan Halaman Ujian Siswa dengan Countdown Timer

Fitur ini membantu memastikan bahwa pengerjaan ujian berlangsung sesuai waktu yang telah ditentukan. Setelah ujian selesai, siswa dapat melihat ringkasan nilai akhir tanpa menampilkan detail kunci jawaban agar keamanan bank soal tetap terjaga.

Implementasi database dilakukan secara relasional menggunakan MySQL dengan sembilan tabel utama, yaitu *classes*, *users*, *teacher_classes*, *subjects*, *questions*, *schedules*, *student_answers*, *results*, dan *clustering_results*. Tabel *classes* digunakan untuk menyimpan data kelas. Tabel *users* digunakan untuk menyimpan akun admin, guru, dan siswa. Tabel *teacher_classes* digunakan untuk memetakan relasi guru dan kelas. Tabel *subjects* digunakan untuk menyimpan mata pelajaran. Tabel *questions* menyimpan bank soal pilihan ganda dan esai. Tabel *schedules* menyimpan jadwal dan sesi ujian. Tabel *student_answers* menyimpan jawaban siswa pada setiap butir soal. Tabel *results* menyimpan rekap nilai akhir siswa. Tabel *clustering_results* menyimpan hasil pengelompokan siswa berdasarkan algoritma K-Means.

Rancangan database memiliki struktur data yang saling berelasi. Relasi antara *users* dan *classes* digunakan untuk menghubungkan siswa dengan kelasnya. Relasi antara *questions*, *subjects*, dan *classes* digunakan untuk memastikan soal terikat pada mata pelajaran dan kelas tertentu. Relasi antara *schedules*, *student_answers*, dan *results* digunakan untuk mencatat proses ujian dari tahap pengerjaan sampai rekapitulasi nilai. Relasi antara *results* dan *clustering_results* digunakan untuk mendukung analisis performa akademik siswa. Desain relasional ini membuat data ujian dapat ditelusuri secara lengkap dari pengguna, soal, jawaban, nilai, sampai hasil klasterisasi.

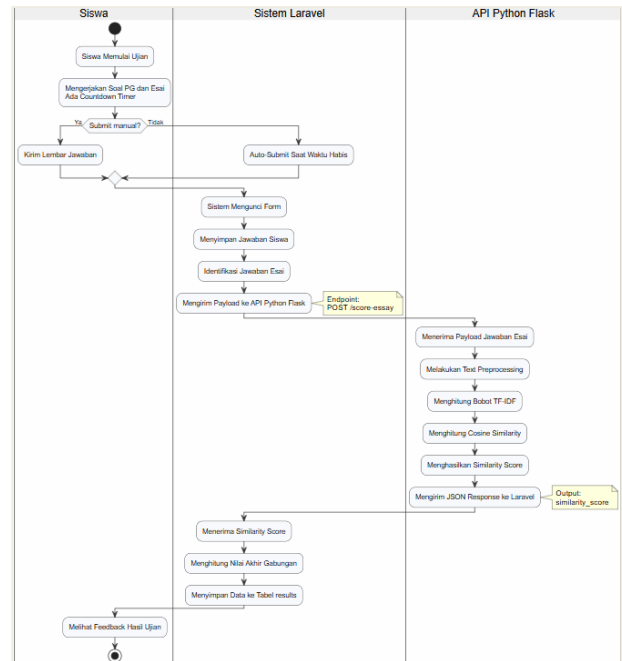
Class diagram sistem menggambarkan model utama dalam Laravel, seperti User, Class, Subject, Question, Schedule, StudentAnswer, Result, dan ClusteringResult. Setiap model memiliki relasi sesuai struktur database.



Gambar 8. Class Diagram Sistem

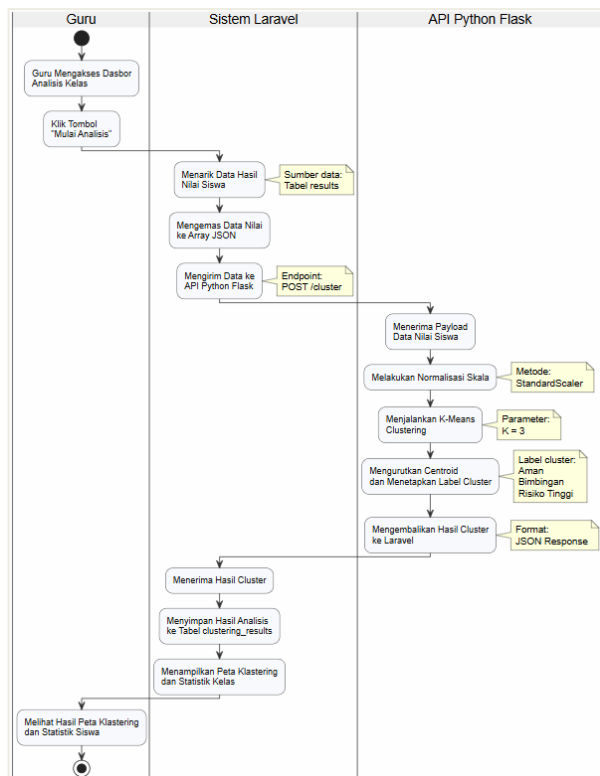
Model *User* berelasi dengan *StudentAnswer*, *Result*, dan *ClusteringResult*. Model *Class* berelasi dengan *User* dan *Schedule*. Model *Subject* berelasi dengan *Question* dan *Schedule*. Model *Schedule* berelasi dengan *StudentAnswer*, *Result*, dan *ClusteringResult*. Relasi ini menunjukkan penerapan pola Model View Controller dalam Laravel, di mana model digunakan untuk mengatur interaksi sistem dengan database.

Activity diagram sistem dibagi menjadi dua alur utama. Alur pertama menggambarkan proses ujian siswa dan penilaian esai berbasis NLP. Proses dimulai ketika siswa mengakses ujian aktif, mengerjakan soal pilihan ganda dan esai, lalu mengirimkan jawaban melalui submit manual atau auto-submit. Laravel menyimpan jawaban siswa, mengidentifikasi jawaban esai, lalu mengirimkan payload ke Flask melalui endpoint penilaian esai. Flask melakukan text preprocessing, pembobotan TF-IDF, dan perhitungan Cosine Similarity. Hasil similarity score dikembalikan ke Laravel untuk dikonversi menjadi skor esai dan digabungkan dengan skor pilihan ganda.



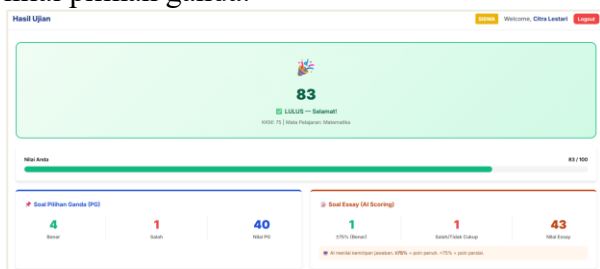
Gambar 9. Activity Diagram Proses Ujian Siswa dan Penilaian Esai

Alur kedua menggambarkan proses analisis performa akademik siswa menggunakan K-Means Clustering. Proses dimulai ketika guru membuka dashboard analisis kelas dan menekan tombol mulai analisis. Laravel mengambil data nilai dari tabel results, kemudian mengemas data nilai akhir, nilai pilihan ganda, dan nilai esai dalam format JSON. Data tersebut dikirim ke Flask melalui endpoint /cluster. Flask melakukan normalisasi data menggunakan StandardScaler, menjalankan algoritma K-Means dengan jumlah kluster tiga, lalu memberi label pada setiap kluster berdasarkan rata-rata nilai akhir. Hasil kluster dikembalikan ke Laravel dan disimpan dalam tabel clustering_results.



Gambar 10. Activity Diagram Analisis Kinerja Menggunakan K-Means

implementasi modul NLP menunjukkan bahwa sistem mampu menilai jawaban esai secara otomatis dan menggabungkannya dengan nilai pilihan ganda.



Gambar 11. Tampilan Hasil Koreksi Esai Otomatis Berbasis NLP

Pada tampilan percobaan hasil ujian, siswa memperoleh nilai akhir 83/100 dan dinyatakan lulus karena melampaui KKM 75 pada mata pelajaran Matematika. Nilai tersebut berasal dari nilai pilihan ganda sebesar 40, dengan rincian 4 jawaban benar dan 1 jawaban salah, serta nilai esai sebesar 43 yang diproses menggunakan fitur AI Scoring.

Pada penilaian esai, sistem membandingkan jawaban_siswa dengan kunci_jawaban melalui tahapan case folding, cleaning, tokenizing, stopword removal, pembobotan TF-IDF, dan perhitungan Cosine Similarity. Hasil pada gambar menunjukkan terdapat 1 jawaban esai

dengan kemiripan $\geq 75\%$ yang dikategorikan benar dan 1 jawaban esai yang belum memenuhi ambang batas. Skor esai tersebut kemudian digabungkan dengan skor pilihan ganda, sehingga sistem dapat menampilkan nilai akhir, status kelulusan, dan ringkasan capaian siswa secara real-time.

Hasil implementasi modul K-Means Clustering menunjukkan bahwa sistem mampu mengelompokkan siswa berdasarkan performa akademik pada mata pelajaran. Data yang digunakan meliputi nilai akhir, nilai pilihan ganda, dan nilai esai, kemudian dinormalisasi menggunakan StandardScaler agar setiap fitur memiliki skala yang seimbang. Setelah itu, sistem menjalankan algoritma K-Means dengan tiga kluster, yaitu Aman, Bimbingan, dan Risiko Tinggi.



Gambar 12. Tampilan Hasil implementasi modul K-Means Clustering

Hasil analisis dari percobaan menunjukkan bahwa terdapat 3 siswa dalam kategori Aman dengan rentang nilai tinggi, yaitu 90,0–100,0. Kategori Bimbingan berisi 1 siswa dengan nilai 73,5, sedangkan kategori Risiko Tinggi berisi 1 siswa dengan nilai 40,0. Hasil ini menunjukkan bahwa sistem tidak hanya menampilkan nilai akhir, tetapi juga memberikan informasi posisi akademik siswa secara otomatis. Guru dapat menggunakan hasil klasterisasi ini sebagai dasar tindak lanjut pembelajaran, seperti pengayaan bagi kategori Aman, pendampingan tambahan bagi kategori Bimbingan, serta remedial atau pembinaan khusus bagi kategori Risiko Tinggi.

3.2. Pembahasan

Hasil implementasi menunjukkan bahwa Laravel dan Python Flask dapat diintegrasikan dalam satu sistem evaluasi pembelajaran. Laravel mengelola autentikasi, bank soal, jadwal, jawaban, nilai, dan tampilan dashboard. Flask menjalankan proses TF-IDF, Cosine Similarity, StandardScaler, dan K-Means. Pertukaran data berbasis JSON memungkinkan proses komputasi dilakukan melalui layanan terpisah tanpa memindahkan seluruh fungsi aplikasi utama.

Arsitektur tersebut memberikan pemisahan tanggung jawab yang jelas. Perubahan pada algoritma NLP dan clustering dapat dilakukan pada layanan Python tanpa mengubah seluruh modul transaksi ujian. Pola ini sejalan dengan penerapan *microservice* pada sistem pembelajaran yang memisahkan fungsi berdasarkan layanan (Dahri et al., 2021). Pemisahan layanan tetap menambah kebutuhan pengelolaan endpoint, penanganan kegagalan koneksi, logging, keamanan API, dan pengujian antarlayanan.

Hasil koreksi esai menunjukkan bahwa sistem dapat memproses jawaban melalui tahap prapengolahan, pembobotan TF-IDF, dan Cosine Similarity. Nilai kemiripan dikonversi menjadi skor sesuai bobot soal. Skor tersebut digabungkan dengan nilai pilihan ganda untuk memperoleh nilai akhir. Pada contoh implementasi, siswa memperoleh nilai akhir 83 dari 100 dan dinyatakan lulus karena nilai tersebut melampaui KKM 75.

Hasil tersebut menunjukkan keberhasilan fungsi komputasi dan integrasi data. Nilai 83 belum dapat digunakan sebagai bukti bahwa skor sistem memiliki akurasi yang sama dengan skor guru. Validasi akurasi memerlukan sejumlah jawaban siswa yang dinilai oleh satu atau beberapa guru. Skor guru dan skor sistem kemudian perlu dibandingkan menggunakan metrik yang sesuai.

Penggunaan TF-IDF dan Cosine Similarity pada penelitian ini sejalan dengan Mi'andri, Siregar dan Utami (2021) yang merepresentasikan jawaban dalam Vector Space Model. Hasil penelitian juga mendukung temuan Permata, Rendika dan Julianty (2025) bahwa Cosine Similarity dapat digunakan untuk mengukur kedekatan jawaban dan kunci jawaban dengan proses yang relatif ringan.

Keunggulan metode terletak pada transparansi perhitungan. Guru dapat memahami bahwa skor diperoleh berdasarkan bobot kata dan kedekatan arah vektor. Proses komputasinya juga sesuai untuk aplikasi sekolah yang tidak selalu memiliki perangkat dengan kemampuan pemrosesan tinggi. Metode ini dapat digunakan sebagai baseline sebelum sistem dikembangkan menggunakan model semantik.

Kelemahan utama terlihat ketika siswa memberikan jawaban yang benar menggunakan

istilah berbeda. TF-IDF tidak memiliki representasi kontekstual yang cukup untuk memahami hubungan sinonim, parafrasa, implikasi, dan urutan informasi. Jawaban yang mengulang banyak kata pada kunci juga dapat memperoleh skor tinggi walaupun penjelasannya kurang tepat. Kondisi tersebut menunjukkan bahwa hasil kemiripan perlu dipadukan dengan batas nilai, mekanisme pemeriksaan guru, dan pengujian terhadap jawaban yang bervariasi.

Penelitian Transformer memberikan gambaran arah pengembangan yang lebih semantik. Pradani & Suadaa (2023) menunjukkan bahwa Semantic Textual Similarity berbasis Transformer mampu menangkap hubungan makna lebih baik daripada pendekatan yang hanya menggunakan kesamaan kata. Amalia et al. (2024) memperlihatkan potensi IndoBERT untuk penilaian dan umpan balik esai bahasa Indonesia. Aliyah et al. (2025) memperkuat representasi kalimat melalui kombinasi IndoBERT dan IndoSBERT.

Model Transformer belum langsung dipilih pada penelitian ini karena tujuan pengembangan diarahkan pada sistem yang ringan dan dapat diintegrasikan dengan aplikasi sekolah. Penggunaan Transformer memerlukan data pelatihan, konfigurasi model, memori, dan waktu inferensi yang lebih besar. Pilihan TF-IDF dan Cosine Similarity dapat dipahami sebagai baseline implementatif. Sistem masih dapat dikembangkan menuju semantic embedding setelah tersedia data jawaban dan skor guru yang cukup.

Hasil implementasi K-Means menunjukkan bahwa nilai akhir, nilai pilihan ganda, dan nilai esai dapat digunakan untuk membentuk tiga kelompok performa. Pada data percobaan, kategori Aman terdiri atas tiga siswa dengan rentang nilai 90–100. Kategori Bimbingan terdiri atas satu siswa dengan nilai 73,5. Kategori Risiko Tinggi terdiri atas satu siswa dengan nilai 40.

Hasil tersebut menunjukkan bahwa sistem dapat mengubah data nilai menjadi informasi kelompok yang lebih mudah dibaca. Guru tidak hanya melihat daftar angka, tetapi juga memperoleh kategori awal untuk menentukan tindak lanjut. Siswa pada kategori Aman dapat diarahkan pada pengayaan. Siswa pada kategori

Bimbingan dapat memperoleh latihan tambahan. Siswa pada kategori Risiko Tinggi dapat diprioritaskan untuk remedial dan pendampingan.

Penerapan tersebut sejalan dengan Dacwanda & Nataliani (2021) yang mengelompokkan siswa berdasarkan nilai pengetahuan dan keterampilan. Hendrastuty (2024) juga menunjukkan bahwa K-Means dapat digunakan untuk menemukan pola pada hasil pembelajaran. Azzahra & Sriani (2025) membuktikan bahwa nilai akademik siswa sekolah menengah dapat dikelompokkan untuk membantu sekolah membaca distribusi kemampuan.

Interpretasi hasil klaster pada penelitian ini perlu dilakukan secara hati-hati. Data percobaan masih berjumlah kecil. Satu perubahan nilai dapat memengaruhi posisi centroid dan keanggotaan klaster. Label Aman, Bimbingan, dan Risiko Tinggi juga tidak berasal dari label yang telah ditentukan sebelumnya. Label diberikan setelah rata-rata nilai setiap klaster diketahui.

K-Means membentuk kelompok berdasarkan kedekatan matematis dan bukan berdasarkan diagnosis pedagogis. Siswa yang berada dalam kategori Risiko Tinggi tidak dapat langsung dianggap memiliki kemampuan rendah secara umum. Nilai dapat dipengaruhi oleh tingkat kesulitan soal, kondisi saat ujian, kehadiran, kesiapan belajar, dan kesesuaian materi. Hasil klaster perlu dipadukan dengan informasi akademik lain dan penilaian guru.

Pemilihan tiga klaster mengikuti kebutuhan kategori pada sistem. Penelitian belum membandingkan hasil untuk beberapa nilai (k). Penggunaan Elbow Method, Silhouette Score, atau Davies-Bouldin Index diperlukan untuk menilai apakah tiga klaster merupakan konfigurasi paling sesuai bagi data yang lebih besar. Evaluasi tersebut penting karena penentuan (k) berdasarkan kebutuhan aplikasi belum selalu sama dengan struktur alami data.

Sistem juga menghasilkan nilai dan kategori performa segera setelah proses komputasi selesai. Fitur ini mendukung konsep learning analytics karena data hasil ujian tidak hanya disimpan, tetapi diolah menjadi informasi yang dapat digunakan guru. Santos & Henriques (2023) menjelaskan bahwa data digital

pembelajaran dapat digunakan untuk mengenali performa siswa lebih awal. Fazil et al. (2024) juga menunjukkan bahwa data akademik dan keterlibatan dapat membentuk representasi performa yang berguna bagi pengajar.

Integrasi hasil analitik dengan tindak lanjut menjadi bagian penting dari sistem. Dai et al. (2025) menegaskan bahwa identifikasi siswa berisiko perlu disertai intervensi dan umpan balik. Kategori yang ditampilkan sistem penelitian ini dapat digunakan sebagai pemicu keputusan guru. Sistem belum memberikan rekomendasi pembelajaran secara otomatis. Jenis remedial, bimbingan, dan pengayaan tetap ditentukan oleh guru.

Klaim real-time pada penelitian ini berkaitan dengan aliran proses pada aplikasi. Jawaban diproses setelah dikirim. Nilai kemudian disimpan dan ditampilkan pada dashboard tanpa perhitungan manual. Hasil klasterisasi juga ditampilkan setelah guru menjalankan analisis. Penelitian belum melakukan benchmark untuk membuktikan waktu respons pada jumlah pengguna yang besar. Istilah real-time lebih tepat dipahami sebagai penyajian langsung setelah proses, bukan sebagai jaminan respons dalam batas milidetik tertentu.

Nilai tambah penelitian terletak pada penyatuan beberapa fungsi yang pada penelitian sebelumnya banyak dikembangkan secara terpisah. Modul NLP memproses jawaban esai. Modul penilaian menggabungkan nilai esai dan pilihan ganda. Modul K-Means membentuk kategori performa. Dashboard menyajikan hasil kepada guru dalam satu alur aplikasi.

Integrasi tersebut membedakan penelitian ini dari studi AES yang hanya menghasilkan skor dan penelitian clustering yang hanya menganalisis data nilai yang telah tersedia. Sistem yang dikembangkan dapat digunakan sejak tahap penyusunan soal, pelaksanaan ujian, pengiriman jawaban, koreksi, perhitungan nilai, sampai pemetaan performa. Kontribusi penelitian berada pada integrasi sistem dan bukan pada penciptaan algoritma NLP atau clustering baru.

3.3 Keterbatasan Penelitian

Penelitian memiliki beberapa keterbatasan. Pertama, TF-IDF dan Cosine Similarity masih

berorientasi pada kesamaan kata. Metode tersebut belum mampu memahami sinonim, parafrasa, negasi, dan konteks kalimat secara mendalam.

Kedua, skor otomatis belum dibandingkan dengan skor guru menggunakan data yang memadai. Nilai yang ditampilkan hanya membuktikan bahwa algoritma dapat dijalankan dan diintegrasikan. Tingkat kesesuaian dengan penilai manusia belum dapat disimpulkan.

Ketiga, data klasterisasi pada percobaan masih terbatas. K-Means sensitif terhadap jumlah data, fitur, skala, jumlah klaster, dan centroid awal. Hasil tiga kelompok belum diuji menggunakan indeks validitas klaster.

Keempat, fitur clustering hanya terdiri atas nilai akhir, nilai pilihan ganda, dan nilai esai. Data kehadiran, keterlambatan pengumpulan, frekuensi percobaan, durasi pengerjaan, dan aktivitas pembelajaran belum digunakan.

Kelima, penelitian dilakukan pada satu konteks sekolah. Generalisasi sistem ke sekolah, mata pelajaran, bentuk soal, dan jumlah pengguna yang berbeda masih memerlukan pengujian lanjutan.

Keenam, kemampuan real-time belum diuji menggunakan response time, throughput, concurrent users, dan load testing. Sistem baru menunjukkan bahwa hasil dapat ditampilkan setelah komputasi selesai tanpa rekapitulasi manual.

Ketujuh, penelitian belum menguji aspek fairness dan kemungkinan perbedaan skor pada gaya bahasa siswa yang berbeda. Hasil penilaian otomatis perlu tetap menyediakan ruang bagi guru untuk melakukan peninjauan dan koreksi.

4. PENUTUP

4.1. Kesimpulan

Penelitian ini berhasil mengembangkan sistem evaluasi pembelajaran berbasis web yang mengintegrasikan Natural Language Processing dan K-Means Clustering. Sistem menggunakan Laravel sebagai aplikasi utama, MySQL sebagai basis data, dan Python Flask sebagai layanan komputasi algoritma. Pertukaran data melalui HTTP Request berbasis JSON memungkinkan proses ujian, penyimpanan jawaban, koreksi esai, perhitungan nilai, dan klasterisasi dijalankan dalam satu alur sistem.

TF-IDF dan Cosine Similarity dapat digunakan untuk menghitung kemiripan antara jawaban siswa dan kunci jawaban. Skor esai yang dihasilkan dapat digabungkan dengan nilai pilihan ganda untuk membentuk nilai akhir dan status kelulusan. Hasil tersebut menunjukkan keberhasilan fungsi sistem dalam melakukan koreksi berbasis kemiripan teks. Penelitian belum membuktikan tingkat kesesuaian skor otomatis dengan skor guru.

K-Means dapat digunakan untuk mengelompokkan siswa berdasarkan nilai akhir, nilai pilihan ganda, dan nilai esai. Hasil klaster diterjemahkan menjadi kategori Aman, Bimbingan, dan Risiko Tinggi. Kategori tersebut memberikan informasi awal yang dapat digunakan guru untuk merencanakan pengayaan, pendampingan, atau remedial.

Esensi temuan penelitian terletak pada integrasi koreksi esai otomatis dan pemetaan performa akademik dalam satu sistem evaluasi berbasis web. Sistem tidak hanya menyimpan nilai, tetapi juga mengolah data menjadi informasi yang mendukung keputusan guru. Kontribusi penelitian berada pada integrasi dan implementasi sistem. Akurasi penilaian, kualitas klaster, dan performa real-time pada skala besar masih perlu diuji lebih lanjut.

4.2. Saran

Pengembangan berikutnya perlu menggunakan dataset jawaban esai yang lebih besar dan beragam. Setiap jawaban sebaiknya dinilai oleh beberapa guru menggunakan rubrik yang sama. Skor guru dapat dibandingkan dengan skor sistem menggunakan Mean Absolute Error, Root Mean Squared Error, korelasi, dan Quadratic Weighted Kappa.

Metode penilaian dapat dikembangkan menggunakan IndoBERT, IndoSBERT, Sentence Transformer, atau semantic embedding. Model semantik dapat digunakan untuk menangkap sinonim dan parafrasa. Pendekatan hibrida juga dapat diterapkan dengan menggabungkan kemiripan semantik, kata kunci, panjang jawaban, struktur kalimat, dan fitur linguistik.

Kualitas clustering perlu diuji menggunakan Elbow Method, Silhouette Score, Davies-Bouldin Index, dan Calinski-Harabasz Index. Beberapa nilai (k) perlu dibandingkan sebelum

kategori performa ditetapkan. Fitur clustering dapat diperluas dengan data kehadiran, waktu pengerjaan, keterlambatan tugas, aktivitas LMS, dan riwayat nilai.

Pengujian sistem perlu mencakup response time, throughput, concurrent users, keamanan API, dan penanganan kegagalan layanan. Pengujian tersebut diperlukan agar klaim real-time dapat didukung oleh ukuran teknis.

Guru perlu diberi fitur untuk meninjau dan mengubah skor otomatis. Sistem juga perlu menyimpan skor awal, skor hasil revisi guru, dan alasan perubahan. Mekanisme tersebut dapat meningkatkan transparansi serta menyediakan data baru untuk pengembangan model pada tahap berikutnya.

5. DAFTAR PUSTAKA

- Aliyah, N. E., Sholikah, R. W., Firdausi, H., Ciptaningtyas, H. T., & Sabilla, I. A. (2025). Enhancing Automated Essay Scoring in Bahasa Indonesia with IndoBERT and IndoSBERT. *2025 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 1–7. <https://doi.org/10.1109/SIML65326.2025.11080721>
- Almalawi, A., Soh, B., Li, A., & Samra, H. (2024). Predictive Models for Educational Purposes: A Systematic Review. *Big Data and Cognitive Computing*, 8(12), 187. <https://doi.org/10.3390/bdcc8120187>
- Amalia, A., Lydia, M. S., Kadir, R. A., Tanjung, F. A. U., Ginting, D. S. B., & Gunawan, D. (2024). Automated Indonesian Essay Scoring and Holistic Feedback Using Bidirectional Encoder Representations for Transformers. *2024 8th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, 96–101. <https://doi.org/10.1109/ELTICOM64085.2024.10864959>
- Azzahra, C., & Sriani, S. (2025). Clustering of High School Students Academic Scores Using K-Means Algorithm. *Journal of Information Systems and Informatics*, 7(1), 572–586. <https://doi.org/10.51519/journalisi.v7i1.1029>
- Dacwanda, D. O., & Nataliani, Y. (2021). Implementasi K-Means Clustering untuk Analisis Nilai Akademik Siswa Berdasarkan Nilai Pengetahuan dan Keterampilan. *AITI*, 18(2), 125–138. <https://doi.org/10.24246/aiti.v18i2.125-138>
- Dahri, F., Taufiq, R., & Prayogi, M. A. (2021). Implementation of Microservices Architecture in Learning Management System E-Course Using Web Service Method. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*.
- Dai, W., Lin, J., Jin, F. J.-Y., Tsai, Y.-S., Srivastava, N., Le Bodic, P., Gašević, D., & Chen, G. (2025). Learning Analytics for Early Identification of At-Risk Students and Feedback Intervention. *Journal of Learning Analytics*, 12(3), 102–125. <https://doi.org/10.18608/jla.2025.8735>
- Faseeh, M., Jaleel, A., Iqbal, N., Ghani, A., Abdusalomov, A., Mehmood, A., & Cho, Y.-I. (2024). Hybrid Approach to Automated Essay Scoring: Integrating Deep Learning Embeddings with Handcrafted Linguistic Features for Improved Accuracy. *Mathematics*, 12(21), 3416. <https://doi.org/10.3390/math12213416>
- Fazil, M., Rísquez, A., & Halpin, C. (2024). A Novel Deep Learning Model for Student Performance Prediction Using Engagement Data. *Journal of Learning Analytics*, 11(2), 23–41. <https://doi.org/10.18608/jla.2024.7985>
- Hendrastuty, N. (2024). Penerapan Data Mining Menggunakan Algoritma {K-Means Clustering} dalam Evaluasi Hasil Pembelajaran Siswa. *Jurnal Ilmiah Informatika Dan Ilmu Komputer (JIMA-ILKOM)*, 3(1), 46–56. <https://doi.org/10.58602/jima-ilkom.v3i1.26>
- Kapoh, H., Melo, O. E., & Kimbal, A. A. (2021). Black Box Testing in Web-based Applications: Case Study-Remedial Application at Manado State Polytechnic. *International Journal of Computer Applications*, 174(12), 21–24. <https://doi.org/10.5120/ijca2021921002>
- Li, S., & Ng, V. (2024). Automated Essay Scoring: Recent Successes and Future

- Directions. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8114–8122. <https://doi.org/10.24963/ijcai.2024/897>
- Maghfiroh, S., & Fatah, Z. (2025). Analisis Data Mining dengan Algoritma K-Means Clustering untuk Menentukan Siswa Berprestasi di MTS Miftahul Ulum Bengkak. *Jurnal Mahasiswa Teknik Informatika*, 4(2). <https://doi.org/10.35473/jamastika.v4i2.4527>
- Maryani, Prabowo, H., Gaol, F. L., & Hidayanto, A. N. (2022). Comparison of the System Development Life Cycle and Prototype Model for Software Engineering. *International Journal of Emerging Technology and Advanced Engineering*, 12(4), 155–162. https://doi.org/10.46338/ijetae0422_19
- Mi'andri, M., Siregar, A. C., & Utami, P. Y. (2021). Sistem Penilaian Ujian Otomatis untuk Soal Esai Menggunakan Metode Vector Space Model. *JUTECH: Journal Education and Technology*, 2(2).
- Misgna, H., On, B.-W., Lee, I., & Choi, G. S. (2025). A Survey on Deep Learning-Based Automated Essay Scoring and Feedback Generation. *Artificial Intelligence Review*, 58(2), 36. <https://doi.org/10.1007/s10462-024-11017-5>
- Ngulube, P., & Ncube, M. M. (2025). Leveraging Learning Analytics to Improve the User Experience of Learning Management Systems in Higher Education Institutions. *Information*, 16(5), 419. <https://doi.org/10.3390/info16050419>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Permata, R., Rendika, & Julianty, L. C. (2025). Towards an Automated Essay Evaluation System NLP Based Text Embeddings and Similarity Metrics. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 16(1). <https://doi.org/10.31849/digitalzone.v16i1.26541>
- Pradani, K. A., & Suadaa, L. H. (2023). Automated Essay Scoring Menggunakan Semantic Textual Similarity Berbasis Transformer untuk Penilaian Ujian Esai. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(6), 1177–1184. <https://doi.org/10.25126/jtiik.2023107338>
- Prastyo, P. H., Tungadi, E., & Zuhdi, S. (2025). Indonesian Automated Essay Scoring: A Comparative Study of Pretrained Transformer Models. *Information Technology Education Journal*, 4(2), 120–130. <https://doi.org/10.59562/intec.v4i2.8069>
- Santos, R. M., & Henriques, R. (2023). Accurate, Timely, and Portable: Course-Agnostic Early Prediction of Student Performance from LMS Logs. *Computers and Education: Artificial Intelligence*, 5, 100175. <https://doi.org/10.1016/j.caeai.2023.100175>
- Sufandi, U. U., Aprijadi, D. A., Dhini, B. F., & Wicaksono, A. (2026). Artificial Intelligence in Essay Scoring: A Comparative Study of Transformer-Based Models. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, 15(1). <https://doi.org/10.23887/janapati.v15i1.103823>
- Susilawati, E., Siregar, E. S., Ekalestari, S., & Harahap, E. S. (2025). Development of a Digital Assessment Model for Automated Short Essay Scoring in Islamic Education Study Program. *Fitrah: Journal of Islamic Education*, 6(1). <https://doi.org/10.53802/fitrah.v6i1.1195>