Jurnal Ilmiah Informatika (JIF)
| ISSN (Print) 2337-8379 | ISSN (Online) 2615-1049 |
Jurnal online di akses di http://ejournal.upbatam.ac.id/index.php/jif

Penerapan Term Frequency-Inverse Document Frequency dan Word Embedding untuk Penilaian Esai Otomatis

Rangga Muhammad Firdaus[1], Andy Victor Pakpahan[2]

[1,2] Program Studi Teknik Informatika, Institut Digital Ekonomi LPKIA, Jl. Soekarno Hatta No. 456, Bandung, 40266, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 26-09-2026
Revisi Akhir: 14-03-2026
Diterbitkan *Online*: 31-03-2026

KATA KUNCI

Penilaian Esai Otomatis
Text Mining
TF-IDF
Word Embedding
Cosine Similarity

KORESPONDENSI

E-mail: abang@lpkia.ac.id

A B S T R A C T

Manual essay grading at the vocational school level is a time-consuming and subjective process. This research implemented and evaluated an automatic essay scoring model using a combination of the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm for word weighting and Word Embedding for semantic meaning analysis. The model was tested using a dataset of 360 essay answers from 36 students at SMK Budi Bakti Ciwidey with train-test split validation. The tuning process on the training data showed that a weighting that prioritized semantic analysis (90% Word Embedding) provided the best performance. In the final testing on 90 test data, the model achieved an excellent Mean Absolute Error (MAE) of 6.80, but with a weak Pearson correlation of 0.12 against the teacher's scores. This research concludes that the proposed model is successful in generating scores that are very close to the teacher's scores (low MAE), but still has limitations in terms of scoring consistency (weak correlation), which is influenced by the quality of the key answers and an imbalanced dataset.

1. PENDAHULUAN

Evaluasi merupakan komponen penting dalam proses pembelajaran yang digunakan untuk mengukur tingkat pemahaman, kemampuan analisis, serta keterampilan berpikir kritis siswa. Salah satu bentuk evaluasi yang banyak digunakan dalam proses pembelajaran adalah soal esai, karena mampu menggambarkan pemahaman konseptual dan kemampuan argumentasi siswa secara lebih mendalam dibandingkan soal pilihan ganda. Menurut penelitian mengenai *Automated Essay Scoring* (AES), penilaian esai memungkinkan pengukuran kemampuan kognitif tingkat tinggi seperti analisis, sintesis, dan evaluasi [1].

Namun demikian, proses penilaian esai secara manual masih menghadapi berbagai permasalahan. Penilaian yang dilakukan oleh pengajar seringkali membutuhkan waktu yang lama, terutama ketika jumlah peserta didik cukup banyak. Selain itu, proses penilaian manual juga berpotensi menimbulkan subjektivitas dan inkonsistensi penilaian, sehingga dapat mempengaruhi objektivitas hasil evaluasi. Permasalahan tersebut juga terjadi di SMK Budi Bakti Ciwidey, dimana proses penilaian esai masih dilakukan secara manual sehingga memerlukan waktu yang cukup lama dan rentan terhadap perbedaan penilaian antar pengajar.

Seiring dengan perkembangan teknologi, berbagai penelitian telah mengembangkan sistem penilaian esai otomatis (*Automated Essay Scoring*) dengan memanfaatkan teknik *Natural Language Processing* (NLP) dan *Text Mining*. Penelitian oleh Landauer, Foltz, dan Laham memperkenalkan metode *Latent Semantic Analysis* (LSA) untuk menilai kesamaan makna dalam teks esai secara otomatis [2]. Pendekatan ini kemudian dikembangkan dalam berbagai penelitian yang memanfaatkan representasi teks berbasis vektor untuk mengukur kemiripan dokumen.

Dalam bidang *text representation*, metode *Term Frequency–Inverse Document Frequency* (TF-IDF) merupakan salah satu teknik yang banyak digunakan untuk merepresentasikan dokumen dalam bentuk numerik berdasarkan frekuensi kata dalam dokumen dan keseluruhan korpus [3]. Metode ini efektif dalam mengekstraksi informasi penting dari teks dan banyak digunakan dalam berbagai aplikasi *text mining* dan *information retrieval*. Namun, pendekatan TF-IDF memiliki keterbatasan karena hanya mempertimbangkan frekuensi kata tanpa memahami hubungan semantik antar kata.

Untuk mengatasi keterbatasan tersebut, penelitian terbaru mulai memanfaatkan *Word Embedding*, yaitu teknik representasi kata dalam bentuk vektor yang mampu menangkap hubungan semantik antar kata dalam ruang vektor berdimensi tinggi [4]. *Word Embedding* memungkinkan sistem memahami kemiripan makna antar kata sehingga dapat meningkatkan kualitas analisis teks. Penelitian oleh Alikaniotis, Yannakoudakis, dan Rei

menunjukkan bahwa penggunaan representasi kata berbasis *embedding* dapat meningkatkan performa sistem penilaian esai otomatis secara signifikan [5].

Meskipun berbagai pendekatan telah dikembangkan, sebagian penelitian masih menggunakan metode berbasis frekuensi kata atau model yang membutuhkan sumber daya komputasi yang tinggi seperti *deep learning*. Hal ini menjadi tantangan dalam implementasi sistem penilaian esai otomatis pada lingkungan pendidikan yang memiliki keterbatasan sumber daya komputasi.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pendekatan kombinasi *Term Frequency–Inverse Document Frequency* (TF-IDF) dan *Word Embedding* untuk merepresentasikan teks jawaban esai siswa. Selanjutnya, *Cosine Similarity* digunakan untuk menghitung tingkat kemiripan antara jawaban siswa dengan kunci jawaban. Pendekatan ini diharapkan mampu menghasilkan sistem penilaian esai otomatis yang lebih efisien, konsisten, dan objektif dibandingkan metode penilaian manual.

2. TINJAUAN PUSTAKA

2.1. Penilaian Esai Otomatis

Evaluasi merupakan proses sistematis untuk mengukur pencapaian tujuan pengajaran, di mana esai menjadi salah satu instrumen penting untuk menilai pemahaman konseptual [6]. Sistem penilaian esai otomatis dikembangkan untuk mengatasi tantangan dalam proses penilaian manual seperti waktu yang lama dan subjektivitas penilai. Pendekatan ini umumnya menggunakan kerangka kerja *Knowledge Discovery in Databases* (KDD) untuk mengekstrak pengetahuan dari data teks [7], [8] .

2.2. *Term Frequency – Inverse Document Frequency*

TF-IDF adalah sebuah metode statistika yang digunakan untuk mengukur tingkat kepentingan sebuah kata dalam sebuah dokumen yang merupakan bagian dari sebuah korpus [9]. Bobotnya merupakan perkalian dari *Term Frequency* (TF), yang mengukur frekuensi kata dalam satu dokumen, dan *Inverse Document Frequency* (IDF), yang mengukur keunikan kata di seluruh korpus. Metode ini efektif untuk mengidentifikasi kata kunci yang penting.

2.3. *Word Embedding*

Word Embedding adalah teknik dalam *Natural Language Processing* (NLP) di mana kata-kata direpresentasikan sebagai vektor numerik dalam ruang multidimensi [10]. Keunggulan utamanya adalah kemampuannya menangkap hubungan semantik, di mana kata dengan makna mirip akan memiliki representasi vektor yang berdekatan. Penelitian ini menggunakan *model pre-trained* berbasis arsitektur *Word2Vec*.

2.4. *Cosine Similarity*

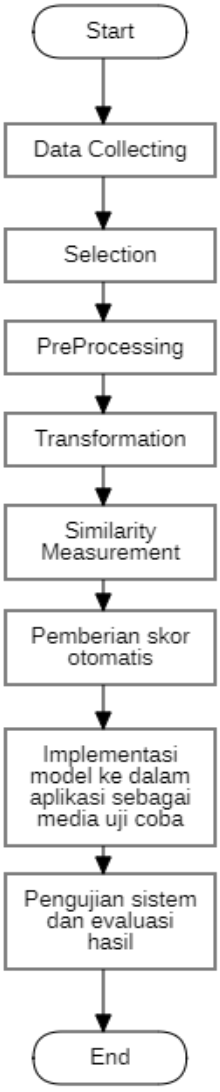
Untuk mengukur kemiripan antara dua teks yang telah direpresentasikan sebagai vektor, digunakan metrik *Cosine Similarity*. Metrik ini bekerja dengan menghitung nilai kosinus dari sudut antara kedua vektor [11]. Hasilnya adalah skor antara 0 dan 1, yang efektif untuk mengukur kemiripan teks tanpa terpengaruh oleh panjangnya.

2.5. Pendekatan Hibrida dan *Ensemble*

Untuk meningkatkan performa, seringkali beberapa model atau metode digabungkan. Salah satu pendekatan yang umum adalah *Ensemble Learning*, di mana beberapa model dasar (*base models*) dikombinasikan untuk menghasilkan prediksi yang lebih kuat. Pakpahan, dkk. [12] telah menunjukkan bahwa teknik *ensemble stacking* dengan mengkombinasikan beberapa model klasifikasi seperti SVM, *Naive Bayes*, dan *Random Forest* dapat meningkatkan akurasi dalam tugas analisis sentimen. Pendekatan penggabungan model seperti ini menjadi dasar pemikiran dalam penelitian ini, di mana dua representasi fitur yang berbeda (TF-IDF dan *Word Embedding*) dikombinasikan untuk tujuan yang sama, yaitu meningkatkan kualitas penilaian.

3. METODOLOGI

Penelitian ini menggunakan desain penelitian kuantitatif dengan pendekatan eksperimental yang dibimbing oleh kerangka kerja *Knowledge Discovery in Databases* (KDD). Alur penelitian terdiri dari beberapa tahapan utama yang akan dijelaskan di bawah ini.



Gambar 1. Alur Penelitian

3.1. *Data Collecting*

Tahap pertama adalah pengumpulan data primer yang terdiri dari tiga komponen utama dari SMK Budi Bakti Ciwidey: (1) 10 set soal esai untuk mata pelajaran Bahasa Indonesia dan PPKn, (2) kunci jawaban yang disusun oleh guru sebagai acuan, dan (3) 360 jawaban siswa yang disertai skor manual sebagai *ground truth*. Seluruh data ini dikompilasi ke dalam satu *file* format CSV.

3.2. *Data Selection*

Tahap selanjutnya adalah seleksi data. Karena data dikumpulkan secara terkontrol, data dipastikan sudah lengkap dan tidak duplikat. Oleh karena itu, langkah seleksi utama yang diimplementasikan adalah penyesuaian struktur data, di mana nama-nama kolom dari *file* CSV distandarkan (misalnya, Jawaban Siswa menjadi jawaban_siswa) untuk memastikan konsistensi dan kemudahan pemrosesan oleh skrip.

3.3. *Preprocessing*

Setelah *dataset* disiapkan, setiap teks jawaban dan kunci jawaban melewati tahap pra-pemrosesan untuk membersihkan dan menstandarkan teks. Proses ini diimplementasikan secara berurutan melalui beberapa langkah: (1) *Case Folding* untuk menyeragamkan huruf menjadi kecil; (2) Penyesuaian Tanda Baca untuk menghapus karakter non-alfabetik; (3) Tokenisasi untuk memecah teks menjadi token kata; (4) *Stopword Removal* untuk menghapus kata-kata umum; dan (5) *Stemming* untuk mengubah kata ke bentuk dasarnya. Teks hasil akhir dari proses ini, yang berupa daftar kata dasar, kemudian digunakan dalam tahap transformasi.

3.4. *Transformation*

Teks bersih dari tahap sebelumnya kemudian ditransformasi menjadi representasi numerik. Proses ini dilakukan dengan dua pendekatan utama. Pertama, pembobotan kata dilakukan menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) yang dilatih pada korpus data latih. Kedua, representasi makna dihasilkan dengan memuat model *Word Embedding pre-trained* (wiki.id.vec). Langkah terakhir adalah mengonversi setiap teks menjadi vektor kalimat tunggal menggunakan metode rata-rata tertimbang yang menggabungkan hasil dari TF-IDF dan *Word Embedding*.

3.5. *Similarity Measurement*

Setelah setiap teks ditransformasi menjadi vektor numerik, dilakukan pengukuran kemiripan antara vektor jawaban siswa dan vektor kunci jawaban. Metode yang digunakan adalah *Cosine Similarity*, yang mengukur sudut antara dua vektor untuk menghasilkan skor kemiripan antara 0 dan 1. Proses ini diterapkan pada kedua representasi vektor, yaitu yang berbasis TF-IDF dan yang berbasis *Word Embedding*.

3.6. Pemberian Skor Otomatis

Tahap terakhir adalah pemberian skor otomatis. Dua nilai kemiripan yang dihasilkan dari tahap sebelumnya dikombinasikan menjadi satu skor akhir dalam skala 0-100. Implementasi dilakukan dengan metode rata-rata tertimbang (*weighted average*), menggunakan resep bobot optimal (10% TF-IDF dan 90% *Word Embedding*) yang ditemukan dari proses *tuning* pada data latih.

3.7. Implementasi Model ke Aplikasi

Untuk memvalidasi alur kerja algoritma secara menyeluruh, sebuah prototipe aplikasi berbasis web dikembangkan. Prototipe ini berfungsi sebagai media untuk pengujian, di mana jawaban esai dapat diinput dan dinilai secara otomatis oleh model yang telah dirancang, membuktikan bahwa model dapat diterapkan dalam sistem nyata.

3.8. Pengujian Sistem dan Evaluasi Hasil

Tahap akhir dari penelitian ini adalah evaluasi kinerja model menggunakan strategi validasi *Train-Test Split* (75% data latih, 25% data uji). Performa model diukur secara kuantitatif dengan membandingkan skor otomatis terhadap skor manual dari guru sebagai *ground truth* menggunakan beberapa metrik: *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), *Pearson Correlation Coefficient*, *Classification Accuracy*, dan Rata-rata Waktu Eksekusi.

4. HASIL DAN PEMBAHASAN

4.1. Pengambilan *Dataset* (*Data Collecting*)

Penelitian ini menggunakan *dataset* primer yang dikumpulkan dari 36 siswa kelas XI jurusan RPL di SMK Budi Bakti Ciwidey. Proses pengumpulan data menghasilkan tiga komponen utama:

1. Dokumen Soal Esai: Sebanyak 10 soal esai (Bahasa Indonesia & PPKn) yang dirancang untuk menguji pemahaman deskriptif dan naratif.
2. Kunci Jawaban Guru: Paragraf kunci jawaban ideal yang disusun oleh guru untuk setiap soal sebagai benchmark.
3. Jawaban Siswa: Sebanyak 360 set jawaban esai otentik, lengkap dengan skor manual dari guru sebagai *ground truth*.

Ketiga komponen ini dikompilasi ke dalam satu file digital jawaban_siswa.csv. Tabel 1 di bawah ini menyajikan sampel dari *dataset* mentah yang telah dikumpulkan.

Tabel 1. Sampel Data Mentah dari file jawaban_siswa.csv

ID Esai	Jawaban Siswa	Kunci Jawaban	Skor Guru
1	Peristiwa Rengasdengklok itu intinya adalah saat para pemuda [...]	Peristiwa Rengasdengklok terjadi ketika golongan muda [...]	88
2	Candi Borobudur adalah candi Hindu yang besar sekali di Jawa Tengah. [...]	Candi Borobudur memiliki struktur punden [...]	55
3	Cerita Malin Kundang itu tentang [...] pesannya kita harus baik sama ibu..	Legenda Malin Kundang mengisahkan seorang [...]	70

4.2. Seleksi *Dataset* (*Data Selection*)

Setelah data mentah dikompilasi, dilakukan tahap penyiapan dataset. Proses ini diimplementasikan dengan menstandarkan nama kolom dari file CSV agar konsisten dan mudah diolah oleh skrip. Tabel 2 di bawah ini mengilustrasikan proses "sebelum" dan "sesudah" dari standardisasi ini.

Tabel 2. Hasil Standardisasi Nama Kolom

Nama Kolom Sebelum (dari CSV)	Nama Kolom Sesudah (di program)
Student Name	student_name
Student NIPD	student_nipd
Jawaban Siswa	jawaban_siswa
Kunci Jawaban	kunci_jawaban
Nilai Guru	nilai_guru

4.3. Implementasi Pra-pemrosesan Teks (*Preprocessing*)

Setelah *dataset* disiapkan, setiap teks jawaban dan kunci jawaban melewati tahap pra-pemrosesan (*preprocessing*) untuk membersihkan dan menstandarkan teks. Proses ini diimplementasikan secara berurutan melalui beberapa langkah: (1) *Case Folding*, (2) Penyesuaian Tanda Baca, (3) Tokenisasi, (4) *Stopword Removal*, dan (5) *Stemming*. Tabel 3 di bawah ini mendemonstrasikan keseluruhan alur proses pra-pemrosesan pada sebuah kalimat contoh.

Tabel 3. Demonstrasi Alur Proses Pra-pemrosesan Teks

No.	Tahap Proses	Contoh Data
1.	Teks Asli (Mentah)	Proses pembelajaran SANGAT penting, bukan?
2.	Sesudah Case Folding	proses pembelajaran sangat penting, bukan?
3.	Sesudah Penyesuaian Tanda Baca	proses pembelajaran sangat penting bukan
4.	Sesudah Tokenisasi	['proses', 'pembelajaran', 'sangat', 'penting', 'bukan']
5.	Sesudah Stopword Removal	['proses', 'pembelajaran', 'penting']
6.	Hasil Akhir (Sesudah *Stemming*)	['proses', 'ajar', 'penting']

4.4. Implementasi Transformasi Data

Teks hasil akhir dari tahap pra-pemrosesan kemudian ditransformasi menjadi representasi numerik. Proses ini dilakukan dengan dua pendekatan: pembobotan kata menggunakan TF-IDF dan representasi makna menggunakan *Word Embedding*.

Pertama, setiap kata diberi bobot kepentingan menggunakan TF-IDF. Kedua, setiap teks diubah menjadi satu vektor kalimat tunggal dengan metode rata-rata tertimbang vektor kata. Tabel 4 di bawah ini mengilustrasikan proses transformasi dari teks bersih menjadi representasi numerik akhir.

Tabel 4. Ilustrasi Proses Transformasi Teks menjadi Vektor

Tahap	Proses dan Ilustrasi
Data "Sebelum"	Teks Bersih: 'candi borobudur waris'
Proses	1. Hitung Bobot TF-IDF per Kata Untuk setiap kata, dihitung bobot kepentingannya. 2. Ambil Vektor Kata (*Word Embedding*) Untuk setiap kata, diambil representasi maknanya dari model *pre-trained*. 3. Hitung Vektor Kalimat (Rata-rata Tertimbang) Vektor setiap kata dikalikan dengan bobot TF-IDF-nya, lalu semua hasilnya dijumlahkan dan dirata-ratakan.
Data "Sesudah"	Vektor Kalimat Final: [0.53, 0.25, -0.11, ...]

4.5. Implementasi Pengukuran Kemiripan dan Pemberian Skor

Setelah setiap teks esai ditransformasi menjadi vektor numerik, tahap selanjutnya adalah mengukur kemiripan. Langkah pertama dalam proses ini adalah menentukan pasangan vektor, yaitu satu vektor untuk jawaban siswa dan satu vektor untuk kunci jawaban. Tabel 5 di bawah ini mengilustrasikan proses "sebelum" dan "sesudah" dari tahap ini, di mana dua buah teks yang berbeda diubah menjadi dua buah vektor numerik yang siap untuk dibandingkan.

Tabel 5. Ilustrasi Proses Penentuan Pasangan Vektor

Data "Sebelum" (Bentuk Teks Bersih)	Proses	Data "Sesudah" (Bentuk Pasangan Vektor)
1. 'candi borobudur waris budaya' 2. 'candi borobudur punya struktur punden'	→ Rata-rata Tertimbang Vektor Kata →	1. Vektor A: [0.45, 0.10, ...] 2. Vektor B: [0.41, 0.12, ...]

Seperti yang ditunjukkan pada tabel 5, sistem berhasil menghasilkan sepasang vektor (Vektor A dan Vektor B) yang siap untuk dihitung tingkat kemiripannya pada tahap selanjutnya. Setelah pasangan vektor (Vektor A dan Vektor B) terbentuk, langkah selanjutnya adalah menghitung nilai kemiripan di antara keduanya menggunakan *Cosine Similarity*, sesuai dengan rumus teoretis. Proses ini mengukur sudut antara dua vektor untuk menghasilkan skor kemiripan antara 0 dan 1. Tabel 6 di bawah ini mengilustrasikan proses perhitungan tersebut secara konseptual.

Tabel 6. Ilustrasi Proses Perhitungan *Manual Cosine Similarity*

Tahap	Proses dan Ilustrasi
Data "Sebelum" (*Input*)	Vektor A: [0.8, 0.4, 0.2] Vektor B: [0.6, 0.5, 0.1]
Proses Perhitungan	1. Hitung *Dot Product* (A · B): Nilai dari setiap elemen yang bersesuaian dikalikan lalu dijumlahkan: = (0.8 × 0.6) + (0.4 × 0.5) + (0.2 × 0.1) = 0.48 + 0.20 + 0.02 = 0.70 2. Menghitung *Magnitude* Vektor A (‖A‖): Setiap elemen dikuadratkan, dijumlahkan, lalu diakarkuadratkan. = sqrt($0.8^2 + 0.4^2 + 0.2^2$) = sqrt(0.64 + 0.16 + 0.04) = sqrt(0.84) ≈ 0.92 3. Menghitung *Magnitude* Vektor B (‖B‖): Proses yang sama diterapkan pada Vektor B. = sqrt($0.6^2 + 0.5^2 + 0.1^2$) = sqrt(0.36 + 0.25 + 0.01) = sqrt(0.62) ≈ 0.79 4. Hitung Skor *Cosine Similarity* = (Hasil Langkah 1) / (Hasil Langkah 2 × Hasil Langkah 3) = 0.70 / (0.92 × 0.79) = 0.70 / 0.7268 ≈ 0.96
Data "Sesudah" (*Output*)	Nilai Kemiripan antara Vektor A dan B adalah 0.96.

Kedua nilai kemiripan yang dihasilkan (satu dari TF-IDF, satu dari *Word Embedding*) kemudian dikombinasikan menggunakan metode rata-rata tertimbang dengan bobot 10% untuk TF-IDF dan 90% untuk *Word Embedding*. Bobot ini merupakan hasil optimal yang ditemukan dari proses *tuning* pada data latih. Hasil akhir dari proses ini adalah satu skor tunggal dalam skala 0-100. Tabel 7 di bawah ini mengilustrasikan proses perhitungan skala 0-100.

Tabel 7. Interpretasi Nilai Kemiripan

Jenis Kemiripan	Nilai (Skala 0-1)	Interpretasi
TF-IDF (kata kunci)	0.3050	Kemiripan kata kunci tergolong rendah.
Word Embedding (makna)	0.9357	Kemiripan makna tergolong sangat tinggi.

4.6. Implementasi Pemberian Skor Otomatis

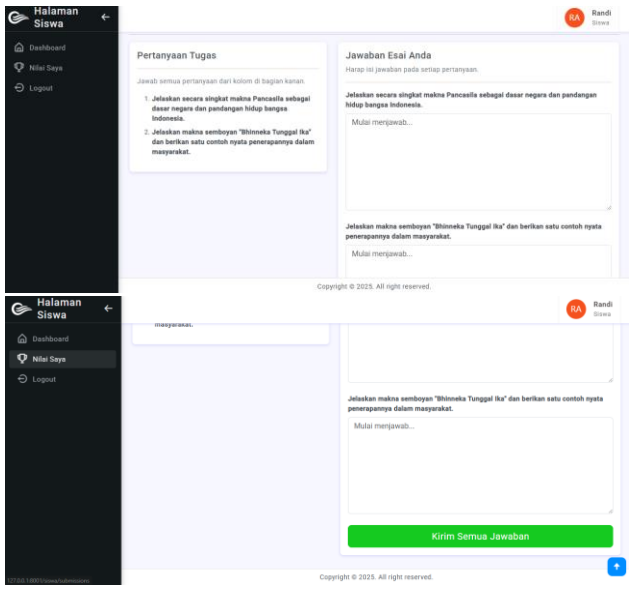
Dua nilai kemiripan yang dihasilkan satu dari TF-IDF dan satu dari *Word Embedding* kemudian dikombinasikan untuk menghasilkan satu skor akhir. Proses ini menggunakan metode rata-rata tertimbang dengan bobot 10% untuk TF-IDF dan 90% untuk *Word Embedding*, yang merupakan hasil dari proses tuning. Tabel 8 di bawah ini mengilustrasikan proses perhitungan manual untuk menghasilkan skor akhir dalam skala 0-100.

Tabel 8. Ilustrasi Proses Perhitungan Skor Akhir

Tahap	Proses dan Perhitungan
Data "Sebelum" (*Input*)	Skor Kemiripan TF-IDF: 0.3050 Skor Kemiripan Semantik: 0.9357
Proses Perhitungan	Skor Akhir = (0.1 × Skor TF-IDF) + (0.9 × Skor Semantik) = (0.1 × 0.3050) + (0.9 × 0.9357) = 0.0305 + 0.84213 = 0.87263
Konversi ke Skala 100	= 0.87263 × 100
Data "Sesudah" (*Output*)	Nilai Akhir Siswa: 87.26

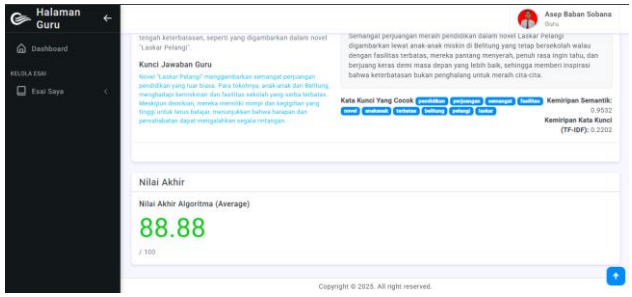
4.7. Implementasi dan Pengujian Prototipe

Sebagai bukti implementasi dan media untuk pengujian, algoritma penilaian esai otomatis yang telah dirancang diintegrasikan ke dalam sebuah prototipe aplikasi berbasis web bernama Esai Otomatis. Antarmuka aplikasi dirancang secara fungsional dan minimalis untuk memfasilitasi alur kerja utama, yaitu proses input jawaban oleh siswa dan penampilan output skor oleh sistem. Gambar 2 di bawah ini menunjukkan antarmuka halaman di mana siswa dapat membaca instruksi dan menuliskan jawaban esai mereka pada kolom yang telah disediakan.



Gambar 2. Tampilan Antarmuka Masukan Jawaban Esai

Setelah siswa mengirimkan jawabannya, aplikasi akan memprosesnya di latar belakang. Guru kemudian dapat melihat hasil penilaian otomatis yang dikeluarkan oleh sistem, seperti yang ditunjukkan pada Gambar 3 Halaman ini menampilkan skor akhir serta detail analisis kemiripan yang dihasilkan oleh algoritma.



Gambar 3. Tampilan Antarmuka Hasil Penilaian Esai

Keberhasilan implementasi prototipe ini menunjukkan bahwa algoritma yang dikembangkan tidak hanya valid secara teoretis dalam lingkungan eksperimen, tetapi juga dapat diimplementasikan dalam sebuah aplikasi nyata.

4.8. Hasil Evaluasi Kinerja Model

Model dengan konfigurasi optimal kemudian dievaluasi pada 90 data uji. Kinerja model diukur menggunakan beberapa metrik kuantitatif. Tabel 9 di bawah ini menyajikan rangkuman hasil evaluasi akhir.

Tabel 9 Hasil Metrik Kuantitaif Model Final

Metrik Evaluasi	Nilai yang Diperoleh
Mean Absolute Error (MAE)	6,80
Root Mean Square Error (RMSE)	8,53
Pearson Correlation Coefficient	0,12
Final System Accuracy	31,11%
Rata-rata Waktu Eksekusi	0.0024 detik/esai

Berdasarkan hasil pada Tabel 9, dapat ditarik beberapa pembahasan utama. Model menunjukkan kinerja yang sangat baik dalam hal kedekatan skor, dibuktikan dengan nilai MAE yang rendah (6.80). Nilai RMSE yang tidak jauh berbeda (8.53) juga mengindikasikan bahwa tidak terdapat banyak kesalahan prediksi yang ekstrem. Selain itu, model terbukti sangat cepat, dengan waktu eksekusi rata-rata hanya 0.0024 detik per esai.

Namun, ditemukan tantangan dalam hal konsistensi penilaian, yang ditunjukkan oleh nilai Korelasi Pearson yang masih lemah (0.12). Rendahnya akurasi klasifikasi *grade* (31.11%) juga menjadi temuan penting, yang setelah dianalisis lebih dalam melalui *Confusion Matrix*, terbukti disebabkan oleh *dataset* yang tidak seimbang, di mana model kesulitan belajar dari *grade* dengan jumlah sampel yang sedikit. Temuan ini menunjukkan bahwa meskipun model akurat secara nilai, konsistensinya dalam meniru pola peringkat guru masih memerlukan perbaikan di masa depan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan dan mengevaluasi model penilaian esai otomatis menggunakan kombinasi TF-IDF dan *Word Embedding*. Hasil penelitian menunjukkan bahwa model yang diusulkan sangat cepat, dengan waktu eksekusi rata-rata hanya 0.0024 detik per esai, dan memiliki tingkat kedekatan skor yang baik dengan penilaian guru (MAE 6.80). Namun, ditemukan tantangan dalam hal konsistensi penilaian, yang ditunjukkan oleh nilai korelasi Pearson yang masih lemah (0.12), serta akurasi klasifikasi *grade* yang rendah akibat *dataset* yang tidak seimbang. Dengan demikian, model ini terbukti berpotensi meringankan beban kerja guru dan mengatasi masalah keterlambatan penilaian, namun konsistensinya masih memerlukan perbaikan. Untuk penelitian selanjutnya, disarankan untuk menggunakan *dataset* yang lebih besar dan seimbang, menguji metode *exemplar-based scoring* untuk meningkatkan konsistensi, serta mengeksplorasi arsitektur *Transformer* seperti IndoBERT untuk pemahaman konteks yang lebih mendalam.

DAFTAR PUSTAKA

[1] M. D. Shermis and J. Burstein, Eds., *Handbook of automated essay evaluation: Current applications and new directions.* New York, NY, US: Routledge/Taylor & Francis Group, 2013.

[2] P. W. Laham, “An Introduction to Latent Semantic Analysis,” 1998.

[3] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Jan. 1988, doi: 10.1016/0306-4573(88)90021-0.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Sep. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781

[5] D. Alikaniotis, H. Yannakoudakis, and M. Rei, “Automatic Text Scoring Using Neural Networks.” [Online]. Available: http://www.kaggle.com/c/asap-aes/

[6] A. Saputra, “Strategi Evaluasi Pembelajaran Pendidikan Agama Islam Pada SMP.”

[7] Y. B. Utomo, I. Kurniasari, and I. Yanuartanti, “PENERAPAN KNOWLEDGE DISCOVERY IN DATABASE UNTUK ANALISA TINGKAT KECELAKAAN LALU LINTAS,” *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 7, no. 1, 2023.

[8] S. Zanki, N. Pusparini, N. Kharisma, and Samuel, “Journal tujuan 1,” Sep. 2025.

[9] H. D. Abubakar and M. Umar, “Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec,” *SLU Journal of Science and Technology*, vol. 4, no. 1 & 2, pp. 27–33, Aug. 2022, doi: 10.56471/slujst.v4i.266.

[10] A. T. Laksana, S. Sylviani, and A. Triska, “STUDI PENERAPAN KONSEP VEKTOR DALAM PERMASALAHAN PENYISIPAN KATA-KATA MELALUI PROSES NORMALISASI VECTOR DAN TRANSFORMASI ORTHOGONAL,” vol. 5, no. 2, 2024, doi: 10.46306/lb.v5i2.

[11] F. Teknik, “PENERAPAN TEKS MINING DAN COSINE SIMILARITY UNTUK MENENTUKAN KESAMAAN DOKUMEN SKRIPSI APPLICATION OF TEXT MINING AND COSINE SIMILARITY TO DETERMINE THE SIMILARITY OF THESIS DOCUMENTS,” 2024.

[12] A. Pakpahan, F. Ferdiansyah, R. Gustian, M. Faiz, and A. Sukma, “Andy Victor,” vol. 7 No. 1 June 2025, Jun. 2025, doi: doi.org/10.35970/jinita.v7i1.2724.

BIODATA PENULIS

Penulis Pertama

Rangga Muhammad Firdaus merupakan mahasiswa di Institut Digital Ekonomi LPKIA, Program Studi Teknik Informatika.

Penulis Kedua
Andy Victor Pakpahan, M.T merupakan Wakil Rektor serta Dosen di Institut Digital Ekonomi LPKIA.