sudo
Jurnal Teknik Informatika
https://jurnal.ilmubersama.com/index.php/sudo

Artikel Penelitian

Sistem Penilaian Esai Otomatis Menggunakan Text Embedding dan Cosine Similarity pada Platform Ujian Daring

Ravi Tegar Al Amin [*], *Shandi Noris*

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 13 Maret 2026
Revisi Akhir: 04 April 2026
Diterbitkan *Online*: 10 April 2026

KATA KUNCI

Automated Essay Scoring
Cosine Similarity
Ujian Daring
Natural Language Processing
Text Embedding

KORESPONDENSI (*)

Phone: +62 851-5921-1502
E-mail: tegaralamin@gmail.com

A B S T R A K

Penilaian esai secara manual memerlukan waktu dan tenaga yang besar, khususnya saat jumlah siswa cukup banyak. MTs Nurul Islam Cisauk masih menggunakan sistem ujian berbasis kertas dengan koreksi manual oleh guru. Penelitian ini merancang dan membangun sistem ujian daring terintegrasi dengan Automated Essay Scoring (AES) menggunakan model text-embedding-3-small dari OpenAI dan algoritma cosine similarity, dikembangkan dengan framework TALL (Tailwind CSS, Alpine.js, Laravel, Livewire). Pendekatan embedding kontekstual berbasis transformer terbukti lebih unggul dalam menangkap makna semantik jawaban siswa dibandingkan metode tradisional seperti TF-IDF, khususnya pada konteks jawaban esai yang beragam di tingkat sekolah menengah. Pengujian dilakukan melalui black box testing, white box testing dengan analisis cyclomatic complexity, serta User Acceptance Testing (UAT) menggunakan kuesioner skala Likert (1–5) terhadap 3 guru dan 7 siswa. Seluruh skenario black box testing bernilai valid. Nilai cyclomatic complexity pada fitur inti berkisar 2–3, menunjukkan alur logika yang sederhana dan terstruktur. Hasil UAT menunjukkan rata-rata kepuasan guru pada rentang 4,00–5,00 dan kepuasan siswa pada rentang 3,86–5,00 dari skala 5,00. Sistem berhasil mengotomasi koreksi esai dan dinilai layak sebagai pendukung evaluasi pembelajaran.

PENDAHULUAN

Evaluasi pembelajaran berbasis esai diakui efektif dalam mengukur kemampuan berpikir kritis siswa, namun prosesnya yang manual membutuhkan waktu dan tenaga yang signifikan. MTs Nurul Islam Cisauk, madrasah tsanawiyah berdiri sejak 1993 di Kecamatan Cisauk, Kabupaten Tangerang, masih menyelenggarakan ujian tulis konvensional berbasis kertas. Kondisi ini menyebabkan inkonsistensi penilaian antar guru, pemborosan sumber daya, dan beban kerja yang meningkat seiring jumlah siswa.

Automated Essay Scoring (AES) merupakan sistem berbasis komputer yang dirancang untuk memberikan skor pada tulisan peserta didik secara otomatis [1]. Penelitian terdahulu oleh Lahitani [2] menerapkan TF-IDF dan cosine similarity pada penilaian esai multi soal dengan hasil objektif, sementara Abdurrahman et al. [3] melaporkan akurasi 75,11% pada implementasi serupa. Namun kedua pendekatan bergantung pada representasi teks statis (TF-IDF) yang tidak mampu menangkap makna kontekstual.

Perkembangan model transformer seperti BERT [4] dan model *text-embedding-3-small* dari OpenAI [5] memungkinkan representasi teks berdimensi tinggi yang menangkap semantik kontekstual. API OpenAI dipilih karena menyediakan dukungan pemrosesan bahasa Indonesia yang memadai secara langsung melalui antarmuka REST tanpa memerlukan infrastruktur GPU lokal, serta kemudahan integrasi yang signifikan dibandingkan model open-source seperti IndoBERT

https://doi.org/10.56211/sudo.v5i1.1655

yang menuntut sumber daya komputasi lokal lebih tinggi dan proses fine-tuning tersendiri. Permata et al. [6] mengonfirmasi cosine similarity lebih efektif dari pendekatan jarak absolut karena hanya memperhitungkan arah vektor, sehingga lebih akurat mengukur kesamaan makna meskipun jawaban berbeda panjang. Integrasi keduanya menjadi pendekatan menjanjikan dalam AES modern [7]. Penelitian ini bertujuan merancang, membangun, dan mengevaluasi sistem penilaian esai otomatis berbasis embedding kontekstual pada platform ujian daring di MTs Nurul Islam Cisauk, guna menggantikan proses koreksi manual yang memakan waktu dan berpotensi menghasilkan penilaian yang tidak konsisten antarguru.

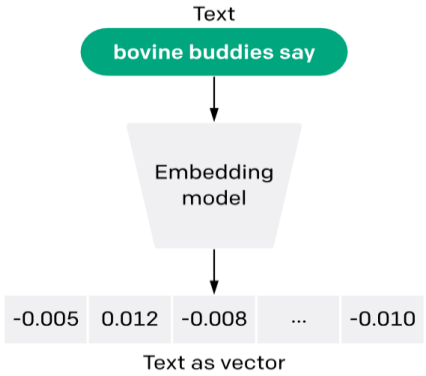
TINJAUAN PUSTAKA

Automated Essay Scoring (AES)

AES merupakan penggunaan teknologi berbasis komputer untuk mengevaluasi dan memberi skor pada tulisan secara otomatis [1]. Sistem AES memproses teks menggunakan berbagai fitur linguistik dan semantik untuk menghasilkan skor yang mendekati penilaian manusia. Pendekatan berbasis kemiripan semantik menjadi salah satu teknik yang banyak dikaji karena fleksibilitasnya dalam menangani variasi ekspresi jawaban [7].

Text Embedding

Text embedding adalah representasi teks dalam bentuk vektor berdimensi tinggi yang menangkap makna semantik suatu kalimat. Model berbasis transformer seperti BERT [4] dan *text-embedding-3-small* dari OpenAI menghasilkan vektor kontekstual yang lebih kaya dibanding metode statis seperti TF-IDF. Albatarni et al. [8] mengonfirmasi bahwa contextual sentence embedding secara konsisten unggul dalam tugas pengukuran kemiripan teks semantik. Ilustrasi proses text embedding ditunjukkan pada Gambar 1.



Gambar 1. Ilustrasi proses text embedding (Sumber: OpenAI [5])

Cosine Similarity

Cosine similarity mengukur kemiripan antara dua vektor berdasarkan sudut di antara keduanya, dengan nilai berkisar antara -1 hingga 1. Keunggulannya dibanding metode jarak absolut adalah ketidakbergantungannya pada panjang vektor, sehingga lebih stabil membandingkan teks dengan jumlah kata berbeda [6]. Rumus cosine similarity ditunjukkan pada persamaan (1).

$$\cos(\theta) = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\cdot\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

A dan B merupakan vektor embedding dari jawaban siswa dan jawaban referensi guru.

Rapid Application Development (RAD)

RAD merupakan model pengembangan perangkat lunak yang menekankan kecepatan pengembangan dan keterlibatan aktif pengguna melalui iterasi bertahap [9]. RAD dipilih karena kesesuaiannya dengan keterbatasan waktu penelitian dan kebutuhan evaluasi langsung dari pengguna di lingkungan sekolah.

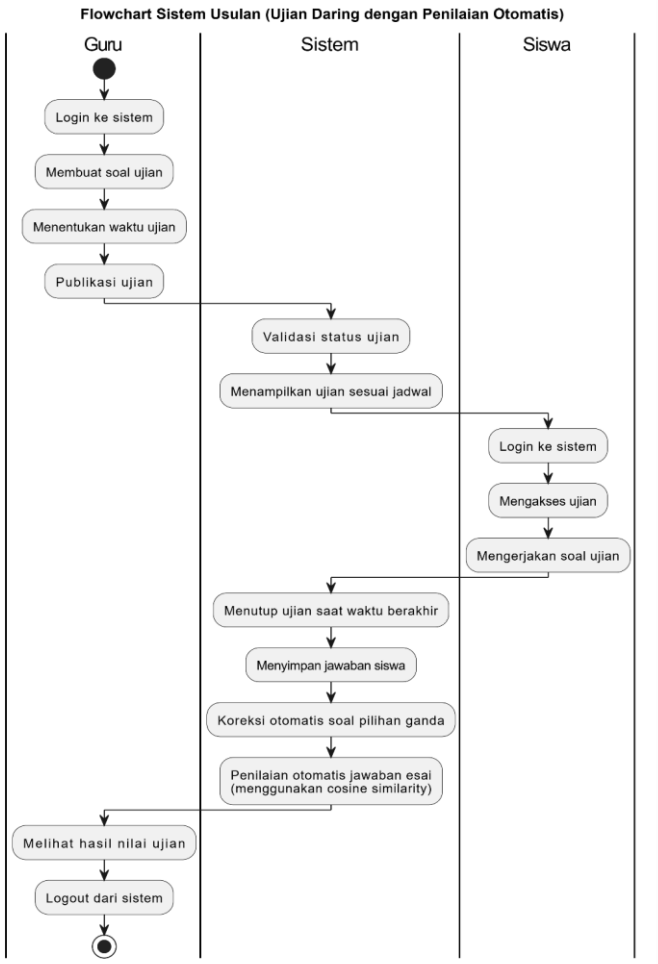
METODOLOGI

Pendekatan dan Model Pengembangan

Penelitian menggunakan pendekatan Research and Development (R&D) dengan model RAD yang terdiri dari empat tahapan: (1) perencanaan kebutuhan melalui studi literatur, observasi, dan wawancara; (2) perancangan cepat mencakup alur sistem, antarmuka, dan basis data; (3) pembangunan dan iterasi melalui implementasi modul ujian daring dan penilaian esai otomatis; serta (4) uji coba dan evaluasi terbatas.

Arsitektur dan Alur Sistem

Sistem dibangun menggunakan framework TALL pada lingkungan PHP 8.2: Tailwind CSS untuk antarmuka, Alpine.js untuk interaktivitas ringan, Laravel 10 sebagai *backend framework*, dan Livewire untuk komponen reaktif. Dari sisi infrastruktur, sistem dikembangkan menggunakan PHP 8.2, MySQL sebagai basis data utama, dan Apache sebagai *web server*. Penilaian esai otomatis diintegrasikan melalui REST API OpenAI menggunakan model *text-embedding-3-small*. Gambar 2 menunjukkan alur sistem secara keseluruhan yang melibatkan tiga aktor: guru, sistem, dan siswa.



Gambar 2. *Flowchart* Sistem Usulan Ujian Daring dengan Penilaian Otomatis

 https://doi.org/10.56211/sudo.v5i1.1655

Mekanisme Penilaian Esai Otomatis

Mekanisme penilaian bekerja dalam tiga tahap: (1) jawaban siswa dan jawaban referensi guru dikirim ke API OpenAI untuk dikonversi menjadi vektor embedding 1536 dimensi; (2) kedua vektor dihitung tingkat kemiripan semantiknya menggunakan cosine similarity sebagaimana ditunjukkan pada Persamaan (1); (3) nilai cos(θ) yang dihasilkan dinormalisasi ke rentang 0–1 kemudian dikalikan dengan bobot soal untuk mendapatkan skor tiap butir essai sebagaimana persamaan (2)

$$\text{Skor}_i = \frac{\cos\theta + 1}{2} \times w_i \tag{2}$$

Normalisasi pada nilai kesamaan dilakukan agar skor selalu positif dan mudah diinterpretasikan. Dengan normalisasi ini, cos(θ) = 1 dipetakan ke skor penuh (bobot soal), cos(θ) = 0 dipetakan ke 0,5 × bobot Soal yang menandakan tidak ada relevansi semantik (vektor ortogonal), dan cos(θ) < 0 mengindikasikan konteks jawaban yang secara semantik bertolak belakang dengan kunci jawaban guru.

Skor yang diperoleh dari Persamaan (2) merupakan skor mentah per butir soal esai. Untuk memperoleh nilai esai yang proporsional terhadap keseluruhan ujian, skor tiap butir diakumulasikan lalu dibandingkan dengan total bobot seluruh soal ujian sebagaimana ditunjukkan pada persamaan (3).

$$\text{Nilai Esai} = \frac{\sum_{i=1}^{n} \text{Skor}_i}{\sum_{j=1}^{m} w_j} \times 100 \tag{3}$$

di mana Skor_i adalah skor ternormalisasi butir soal esai ke-*I* dan w_j adalah bobot soal ke-*j*. Bobot yang dikalkulasikan pada persamaan (3) adalah bobot dari seluruh soal. Nilai yang dihasilkan merupakan kontribusi esai terhadap nilai akhir ujian dalam skala relatif, sehingga bobot esai tetap proporsional meski jumlah soal esai berbeda antar ujian.

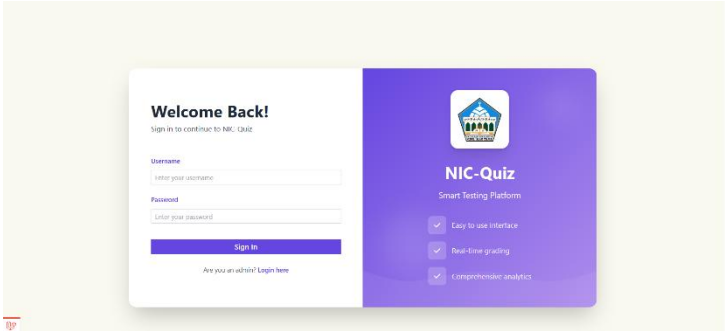
Sampel dan Instrumen Pengujian

Sampel dipilih menggunakan teknik *purposive sampling* berdasarkan keterlibatan langsung dalam uji coba sistem, terdiri dari 3 guru (IPS/kepala madrasah, Bahasa Inggris, IPA) dan 7 siswa. Pengujian dilakukan melalui black box testing untuk memvalidasi fungsionalitas modul krusial, white box testing menggunakan analisis flowgraph dan cyclomatic complexity V(G) = E - N + 2, dan (UAT menggunakan kuesioner skala Likert (1-5) mencakup aspek UI/UX, fungsionalitas, dan kebermanfaatan.

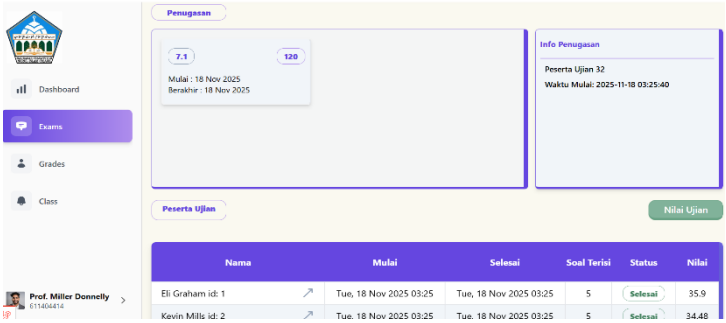
HASIL DAN PEMBAHASAN

Implementasi Sistem

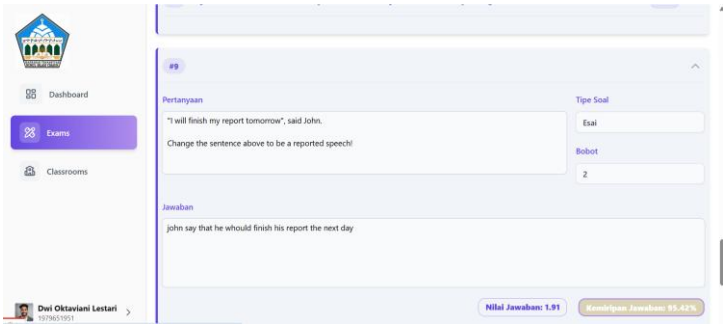
Sistem ujian daring berhasil diimplementasikan dengan nama *NIC-Quiz* dan mendukung tiga peran pengguna: administrator, guru, dan siswa. Gambar 4 menunjukkan tampilan login, Gambar 5 memperlihatkan tampilan hasil ujian peserta yang menampilkan nilai per siswa secara langsung setelah koreksi otomatis dijalankan, sementara Gambar 6 memperlihatkan hasil penilaian per butir soal tiap siswa.



Gambar 4. Tampilan Halaman Login Aplikasi Nic-Quiz



Gambar 5. Tampilan Halaman Hasil Ujian Peserta dengan Nilai Otomatis



Gambar 6. Tampilan Halaman Hasil Ujian Peserta Per Butir Soal

Fitur inti sistem adalah penilaian esai otomatis. Saat guru menekan tombol koreksi, sistem mengambil seluruh jawaban esai peserta, mengirim setiap pasang jawaban ke API OpenAI untuk mendapatkan vektor *embedding* 1536 dimensi, menghitung cosine similarity, dan menyimpan hasilnya sebagai skor. Seluruh proses berlangsung dalam satu iterasi untuk semua peserta melalui mekanisme batching.

Hasil Black Box Testing

Black box testing dilakukan pada enam modul krusial. Seluruh skenario menghasilkan output yang sesuai ekspektasi sebagaimana dirangkum pada Tabel 1.

Tabel 1. Rekapitulasi Hasil *Black Box Testing*

No	Modul	Skenario Utama	Hasil
1	Login	Kredensial valid -> masuk; tidak valid -> ditolak	Valid
2	Tambah Ujian	Data lengkap -> tersimpan; tidak lengkap -> ditolak	Valid
3	Penugasan Ujian	Waktu valid -> tersimpan; waktu konflik -> ditolak	Valid
4	Sunting Soal	Edit soal esai dan pilihan ganda berhasil disimpan	Valid

 https://doi.org/10.56211/sudo.v5i1.1655

5	Koreksi Otomatis	Embedding diproses, cosine similarity dihitung, skor tersimpan	Valid
6	Koreksi Manual	Guru ubah nilai esai, perubahan tersimpan ke database	Valid

Hasil White Box Testing

Pengujian *white box* dilakukan pada tiga fitur inti menggunakan analisis flowgraph dan perhitungan cyclomatic complexity. Hasil disajikan pada Tabel 2. Nilai V(G) <= 3 pada seluruh fitur mengindikasikan struktur logika program yang sederhana dan mudah dipelihara [9].

Tabel 2. Hasil *White Box Testing* dan *Cyclomatic Complexity*

Fitur	N	E	P	V(G)	Jalur Independen
Penugasan Ujian	5	5	1	2	2 jalur, semua valid
Pelaksanaan Ujian	5	5	1	2	2 jalur, semua valid
Koreksi Esai Otomatis	8	9	2	3	3 jalur, semua valid

**N=Node, E=Edge, P=Predicate, V(G)=Cyclomatic Complexity*

Hasil User Acceptance Testing (UAT)

UAT dilaksanakan menggunakan kuesioner skala Likert (1-5) kepada 3 guru dan 7 siswa. Tabel 3 menyajikan hasil kuesioner guru dan Tabel 4 merangkum kuesioner siswa.

Tabel 3. Hasil Kuesioner Guru (n=3)

No	Pernyataan	G1	G2	G3	Rata-rata
1	Tampilan antarmuka mudah dipahami	4	5	5	4.67
2	Pembuatan dan pengelolaan ujian mudah dilakukan	3	5	4	4.00
3	Sistem membantu mengelola ujian secara efisien	5	5	5	5.00
4	Hasil ujian ditampilkan jelas dan informatif	5	5	5	5.00
5	Penilaian otomatis mengurangi beban koreksi	5	5	5	5.00
6	Hasil penilaian otomatis sesuai dengan penilaian guru	4	4	5	4.33
7	Sistem mempercepat proses penilaian ujian	5	5	5	5.00
8	Sistem meningkatkan efisiensi waktu guru	5	5	5	5.00
9	Saya merasa puas menggunakan sistem ini	4	5	4	4.33
10	Sistem layak digunakan dalam ujian di sekolah	5	5	5	5.00
Rata-rata Keseluruhan					**4.70**

**G1=Guru IPS/Kepsek, G2=Guru Bahasa Inggris, G3=Guru IPA*

Tabel 4. Rekapitulasi Hasil Kuesioner Siswa (n=7)

No	Pernyataan	Rentang Skor (S1-S7)	Rata-rata
1	Tampilan antarmuka mudah dipahami	3 - 5	4.57
2	Navigasi menu mudah digunakan	3 - 5	4.57
3	Sistem dapat diakses dengan baik selama ujian	3 - 5	4.29
4	Proses pengerjaan soal berjalan lancar	3 - 5	3.86

5	Informasi waktu ditampilkan dengan jelas	4 - 5	4.86
6	Sistem membantu mengerjakan ujian lebih teratur	3 - 5	4.29
7	Sistem lebih praktis dari ujian manual	3 - 5	4.43
8	Sistem memudahkan mengikuti ujian di kelas	3 - 5	4.29
9	Saya merasa puas menggunakan sistem ini	3 - 5	4.14
10	Sistem layak digunakan sebagai media ujian	5 - 5	5.00

Pembahasan

Rata-rata kepuasan guru sebesar 4,70/5,00 merupakan indikator kuat bahwa integrasi model *text-embedding-3-small* dengan cosine similarity menghasilkan penilaian esai otomatis yang dapat diterima pengguna. Hasil ini melampaui akurasi 75,11% yang dilaporkan Abdurrahman et al. [3] menggunakan TF-IDF statis, mengonfirmasi keunggulan representasi kontekstual transformer sebagaimana ditegaskan Li & Ng [7] dan Albatarni et al. [8].

Meskipun secara umum penilaian otomatis ini mendapat respons positif dengan skor kesesuaian sebesar 4,33 dari para guru, masih ditemukan beberapa anomali yang memperlihatkan kesenjangan signifikan antara skor dari sistem dan koreksi manual guru. Kondisi ini membuktikan bahwa pendekatan *semantic embedding* masih memiliki keterbatasan fundamental. Sebagai contoh, pada salah satu instrumen soal Bahasa Inggris yang menginstruksikan siswa untuk mengubah kalimat menjadi *reported speech* (*"I will finish my report tomorrow", said John.*), ditemukan perbedaan nilai yang sangat ekstrem antara evaluasi sistem dan penilaian guru.

Tabel 5. Perbandingan Hasil Penilaian Sistem dengan Guru

Jawaban Referensi	**Jawaban Murid**	**Penilaian Sistem**	**Penilaiaun Guru**
John said that he would finish his report the next day.	john say that he whould finish his report the next day	0.954	0
	Joni say that He whould finish his report the next day	0.8875	0
	John said that he would finish his report the next day	1	1
	i dont know	0.544	0
	report finish my will tomorrow	0.748	0
	report fnish my will tomorrow	0.7195	0

Berdasarkan Tabel 5, terlihat jelas bahwa model *embedding* hanya menangkap kesamaan konsep atau kedekatan klaster makna dari setiap kata yang digunakan, tanpa memperhatikan struktur tata bahasa (gramatika) maupun keakuratan leksikal. Pada kasus pertama, jawaban siswa memiliki kesalahan *grammar* fatal (*"say"* alih-alih *"said"*) dan kesalahan ejaan/*typo* ekstrem (*"whould"* alih-alih *"would"*), yang bagi guru Bahasa Inggris bernilai mutlak 0. Namun, karena kata-kata tersebut secara semantik berada dalam ruang vektor yang sangat dekat dengan kata aslinya, sistem menganggapnya hampir identik dan memberikan skor 95,4%.

Limitasi semantik ini semakin terbukti pada fenomena *word-salad* (susunan kata acak). Ketika siswa menjawab *"report finish my will tomorrow"*, susunan kalimat tersebut tidak memiliki makna logis secara bahasa. Akan tetapi, karena mesin hanya menghitung *magnitude* dari vektor kata *"report"*, *"finish"*, dan *"tomorrow"* yang juga terdapat pada kunci jawaban, *cosine similarity* gagal mendeteksi kerusakan sintaksis dan tetap memberikan skor kemiripan yang cukup tinggi (74,8%). Kasus *"i dont know"* (skor 54,4%) juga menunjukkan bahwa pendekatan semantik tidak mampu mengklasifikasikan frasa keputusasaan atau pengabaian soal, karena model secara *default* akan selalu mencoba mencari derajat kedekatan sekecil apa pun antara dua teks yang diperbandingkan.

Nilai cyclomatic complexity yang rendah (2-3) pada seluruh fitur inti mengindikasikan arsitektur kode yang sederhana dan mudah dipelihara. Secara khusus, nilai V(G) di kisaran 2–3 berarti setiap fungsi hanya memiliki sedikit jalur eksekusi independen, sehingga meminimalisir potensi bug tersembunyi akibat cabang logika yang kompleks dan memungkinkan

pengujian menyeluruh dengan jumlah test case yang efisien. Ini membuktikan keberhasilan pendekatan RAD dalam menghasilkan sistem fungsional dengan struktur yang terkelola [9]. Skor siswa terendah (3,86) pada kelancaran pengerjaan soal kemungkinan disebabkan keterbatasan infrastruktur jaringan WiFi lokal selama uji coba, bukan kelemahan sistem itu sendiri.

Penelitian ini memiliki keterbatasan yang perlu diakui: sampel terbatas (3 guru, 7 siswa) tidak memungkinkan generalisasi statistik inferensial; sistem bergantung pada API OpenAI berbayar per token; dan penilaian hanya mencakup dimensi semantik tanpa aspek linguistik lainnya. Keterbatasan ini membuka ruang eksplorasi model embedding Bahasa Indonesia berbasis sumber terbuka seperti IndoBERT [10] pada penelitian selanjutnya.

KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan membangun sistem ujian daring terintegrasi dengan penilaian esai otomatis menggunakan model *text-embedding-3-small* dari OpenAI dan algoritma cosine similarity pada framework TALL. Tiga kesimpulan utama: (1) sistem berhasil mempercepat koreksi esai secara signifikan melalui satu tombol koreksi untuk seluruh peserta; (2) sistem menunjukkan konsistensi pengukuran semantik yang dapat diterima dengan rata-rata kepuasan guru 4,70/5,00; (3) seluruh pengujian fungsional menghasilkan hasil valid dengan cyclomatic complexity rendah (2-3), menandakan kualitas implementasi yang baik.

Untuk pengembangan selanjutnya disarankan: (1) eksplorasi model embedding Bahasa Indonesia berbasis sumber terbuka untuk mengurangi ketergantungan API berbayar; (2) penambahan mekanisme umpan balik kepada siswa mengenai kualitas jawaban; (3) perluasan dataset uji coba dengan jumlah siswa yang lebih representatif untuk memungkinkan validasi statistik yang lebih kuat.

DAFTAR PUSTAKA

[1] S. Dikli, "An overview of automated scoring of essays," J. Technol. Learn. Assess., vol. 5, no. 1, pp. 1-35, 2006.
[2] A. Lahitani, "Automated essay scoring menggunakan cosine similarity pada penilaian esai multi soal," J. Teknologi Informasi, vol. 18, no. 2, pp. 145-158, 2022.
[3] A. Abdurrahman et al., "Rancang bangun aplikasi penilaian esai otomatis menggunakan algoritma cosine similarity (Studi kasus MTS Perguruan Muallimat)," Jurnal Informatika, 2021.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171-4186.
[5] OpenAI, "Embeddings," 2023. [Online]. Available: https://platform.openai.com/docs/guides/embeddings
[6] A. R. Permata, Y. Nugroho, and B. Santoso, "Perbandingan cosine similarity dan semantic distance dalam pengukuran kesamaan teks," J. Teknologi Informasi dan Ilmu Komputer, vol. 12, no. 1, pp. 45-54, 2025.
[7] H. Li and V. Ng, "Recent advances in automated essay scoring," ACM Comput. Surv., vol. 56, no. 8, Article 174, 2024.
[8] R. Albatarni, M. H. Alsharif, and S. S. Alotaibi, "Contextual sentence embeddings for semantic text similarity," Inf. Process. Manage., vol. 61, no. 2, Article 103190, 2024.
[9] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York: McGraw-Hill, 2020.
[10] Politeknik Statistika STIS, "AES berbasis semantic text similarity dengan fine-tuned IndoBERT dan cosine similarity," J. Komputasi Statistik, vol. 12, no. 1, pp. 78-92, 2023.

NOMENKLATUR

A	:	Vektor embedding jawaban siswa
B	:	Vektor embedding jawaban referensi guru
cos(θ)	:	Nilai cosine similarity antara vektor A dan B
V(G)	:	Cyclomatic complexity
N	:	Jumlah node pada flowgraph

skor	:	Skor butir soal
w	:	Bobot soal

 https://doi.org/10.56211/sudo.v5i1.1655