

# Kombinasi Natural Language Processing Dengan Generalized Latent Semantic Analysis Pada Deteksi Otomatis E-Learning

Ery Setiyawan Jullev Atmadji<sup>1\*</sup>, Mohammad Arif Khoiruman<sup>2</sup>, Bekti Maryuni Susanto<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknologi Informasi, Politeknik Negeri Jember, Indonesia

<sup>1</sup>ery@polije.ac.id, <sup>2</sup>arif@polije.ac.id, <sup>3</sup>bekti@polije.ac.id

## Abstract

Assessment is an essential element of the educational process, with tests functioning as a prevalent means to gauge students' comprehension of the subject presented. Technological improvements have enabled the administration of tests online via e-learning platforms, offering students prompt feedback on their performance. This technology mostly applies to multiple-choice questions, but essay-type questions need human grading owing to the significant variety in replies. This research seeks to create an automated essay evaluation system by using the Generalized Latent Semantic Analysis (GLSA) algorithm with a thesaurus dictionary. GLSA is used to examine semantic links between phrases, whilst the thesaurus facilitates the identification of synonymous words, hence enhancing the precision of semantic similarity detection. Before semantic analysis, the system employs the Levenshtein Distance technique to identify and rectify typographical problems at the character level. The ultimate similarity between student replies and reference answers is quantified by cosine similarity, yielding a score that indicates the extent of semantic congruence. This suggested method offers a viable option for automating the examination of essay-type questions in e-learning contexts, hence enhancing efficiency and uniformity in educational assessments.

**Keywords:** GLSA, Levenshtein Distance, Thesaurus, Essay-Type Questions, Assessment

## Abstrak

Evaluasi pembelajaran merupakan komponen penting dalam dunia pendidikan, dengan ujian sebagai metode yang umum digunakan untuk mengukur penguasaan materi oleh peserta didik. Perkembangan teknologi telah memungkinkan pelaksanaan ujian secara daring melalui sistem e-learning, yang memberikan hasil evaluasi secara instan. Namun, sistem ini umumnya hanya efektif untuk soal pilihan ganda, sementara untuk soal esai masih memerlukan koreksi manual akibat variasi jawaban yang tinggi, sehingga memerlukan waktu dan tenaga yang lebih besar. Penelitian ini bertujuan untuk merancang dan membangun sistem koreksi otomatis untuk soal esai dengan mengimplementasikan algoritma Generalized Latent Semantic Analysis (GLSA) yang dikombinasikan dengan kamus Thesaurus. GLSA digunakan untuk mengidentifikasi hubungan semantik antar kalimat, sedangkan Thesaurus berperan dalam pencarian kata-kata sinonim guna meningkatkan akurasi kesamaan makna. Sebelum proses penilaian dilakukan, data terlebih dahulu diproses menggunakan algoritma Levenshtein Distance untuk mengenali dan mengoreksi kesalahan pengetikan pada tingkat karakter. Penilaian akhir terhadap kesamaan jawaban peserta ujian dan kunci jawaban dilakukan menggunakan metode cosine similarity untuk menghasilkan skor yang merepresentasikan tingkat kesamaan semantik. Sistem ini diharapkan dapat menjadi solusi efektif dalam mengotomatisasi koreksi soal esai dalam platform e-learning.

**Kata Kunci:** GLSA, Levenshtein Distance, Pengujian Esay, Thesaurus, Pengujian

Published Online 03-05-2025

## I. PENDAHULUAN

Pendidikan merupakan aspek fundamental dalam kehidupan manusia yang berperan penting dalam meningkatkan kualitas sumber daya manusia. Pendidikan mencakup berbagai disiplin ilmu, seperti fisika, sains, matematika, dan bidang lainnya [1]. Untuk memperoleh pendidikan, seseorang dapat menempuh berbagai jenjang pendidikan formal, mulai dari Sekolah Dasar (SD), Sekolah Menengah Pertama (SMP), Sekolah Menengah Atas (SMA) atau Sekolah Menengah Kejuruan (SMK),

hingga jenjang Perguruan Tinggi. Evaluasi terhadap pemahaman dan penguasaan materi pembelajaran dilakukan melalui berbagai metode, salah satunya adalah ujian [2].

Tradisionalnya, ujian dilaksanakan dalam bentuk tertulis, namun, seiring dengan perkembangan teknologi, sistem e-learning telah menjadi alternatif yang banyak digunakan dalam proses evaluasi pembelajaran [3]. Salah satu keunggulan utama dari sistem e-learning adalah kemampuannya dalam memberikan hasil ujian secara langsung, terutama untuk soal pilihan ganda yang memiliki jawaban pasti. Kemudahan ini membuat sistem e-learning semakin diadopsi dalam berbagai institusi Pendidikan [4][5].

Namun, meskipun sistem e-learning memberikan kemudahan dalam penilaian soal pilihan ganda, masih terdapat keterbatasan dalam mengoreksi soal berbentuk esai. Pada tingkat pendidikan yang lebih tinggi, seperti di perguruan tinggi, soal esai lebih sering digunakan karena mampu mengukur kemampuan analisis, sintesis, dan pemahaman konsep secara mendalam. Sayangnya, sistem e-learning konvensional umumnya belum mampu memberikan penilaian otomatis terhadap jawaban esai, karena jawaban pada soal esai bersifat subjektif dan memerlukan penilaian berbasis konteks yang lebih kompleks [6].

Berdasarkan permasalahan tersebut, peneliti akan melakukan Analisa terhadap kinerja dari *Generalized Latent Semantic Analysis* (GLSA) yang digabungkan dengan kamus *Thesaurus* pada koreksi esai otomatis. Untuk kamus *Thesaurus* sendiri digunakan untuk mencari kata serupa sehingga mendapatkan kesamaan data yang lebih akurat. Sebelum diukur hubungan antar kalimat, diproses terlebih dahulu menggunakan *Levenshtein Distance* yang merupakan algoritma pengambilan keputusan untuk mengukur nilai kesamaan dari dua buah kata jika terjadi kesalahan dalam pengetikan. Kemudian diukur kesamaan data dari jawaban yang diberikan pengajar dengan jawaban peserta ujian menggunakan cosine similarity.

## II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Text Mining yang digabungkan dengan GLSA untuk menentukan alternatif takjil terbaik berdasarkan beberapa kriteria. Langkah-langkah dalam metode ini meliputi penentuan Dataset, model NLP hingga proses pemahaman terkait maksud dari sebuah kata dengan pendekatan Thesaurus.

### A. Text Mining

Text mining memberikan sebuah set metodologi dan tool untuk menemukan, memvisualkan, mengevaluasi pengetahuan dari kumpulan besar dari teks dokumen. [7] mengatakan bahwa text mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Text mining, bisa disebut data text mining atau setara dengan analisis teks adalah suatu proses yang digunakan untuk mendapatkan informasi berkualitas tinggi dari suatu teks. Informasi yang didapatkan karena memperhatikan pola dan tren melalui cara seperti mempelajari pola statistic [8].

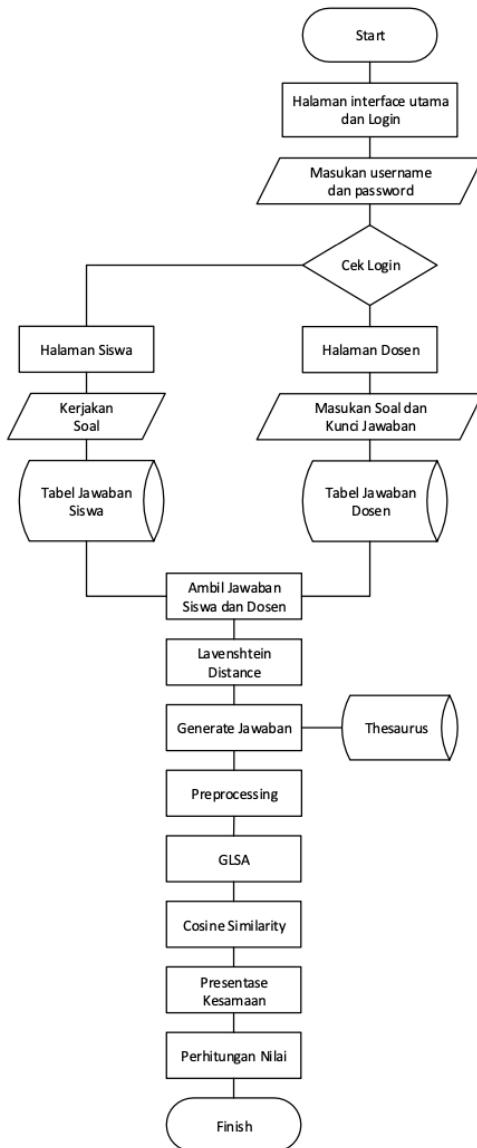
### B. Generalized Latent Semantic Analysis

Ghate mengatakan bahwa Generalized Latent Semantic Analysis (GLSA) merupakan pengembangan lebih lanjut dari algoritma LSA dengan membuat N-Gram berdasarkan matriks dokumen. GLSA mempertimbangkan urutan kata kalimat dalam dokumen dan menjaga kedekatan kata dalam kalimat. Seperti contoh misalnya dalam LSA untuk formasi kata “Kabupaten Jember” menghasilkan arti yang sama dengan “Jember Kabupaten”.

Menurut GLSA, bigram vektor untuk “Kabupaten Jember” itu mempunyai arti yang satu daripada kombinasi “Kabupaten” dan “Jember”. GLSA sendiri tidak berbasis pada vektor dokumen dengan pendekatan bag of words [9]. Melainkan proses dimulai dengan kesamaan-kesamaan antar kata yang berpasangan secara semantik untuk melakukan komputasi representasi untuk kata [10].

### C. Desain Algoritma Sistem Koreksi Otomatis

Desain algoritma sistem merupakan alur tahapan pada sistem dalam melakukan koreksi jawaban pada soal esai. Tahapan ini dimulai dari login mahasiswa atau dosen kemudian dosen memasukkan soal dan kunci jawaban serta mahasiswa memasukkan jawaban. Tahap selanjutnya akan dilakukan pengolahan kata dengan algoritma text minning meliputi Text Preprocessing, Lvenshtein Distance dan Generalized Latent Semantic Analysis. Dari proses tersebut akan menghasilkan bobot dari jawaban milik mahasiswa yang digunakan sebagai acuan sistem dalam menghitung nilai.:.



Gambar 1. Flowchart Aplikasi yang akan dikembangkan

## III. HASIL DAN PEMBAHASAN

Rancangan pembangunan system ini dibuat berdasarkan kebutuhan umum yang ada pada sistem E-Learning serta dengan meninjau sistem E-Learning yang telah ada sebelumnya. Kebutuhan umum tersebut meliputi:

- Dosen dapat menambahkan materi pembelajaran serta menambahkan kuis.
- Dosen mengetahui nilai dari setiap mahasiswa yang telah mengerjakan kuis.
- Mahasiswa mengunduh materi serta mengerjakan kuis yang diberikan dosen.

Sistem ini dibuat berbasis web dengan menggunakan framework CodeIgniter. Untuk mempermudah pembangunan system dibuat rancangan system tersebut meliputi hak akses, kebutuhan

sistem, rancangan alur pada masing hak akses, rancangan sistem dalam melakukan koreksi jawaban serta rancangan database. Sebagai contoh pada implementasi pada GLSA adalah sebagai berikut : Apa yang dimaksud kuartil?

**Kunci Jawaban:**

Nilai yang membagi data yang telah diurutkan kedalam empat bagian yang sama besar.

**Jawaban Siswa:**

Membagi data menjadi empat bagian sama besar. Setelah didapatkan data tersebut pada Tahap awal ditentukan terlebih dahulu query dari kunci jawaban dan jawaban mahasiswa. Query dari kunci jawaban (D) dan query dari jawaban mahasiswa (Qn).

Tabel 1. Pembentukan dokumen Term

Dokumen	Term
D	Nilai yang membagi data yang telah diurutkan kedalam empat bagian yang sama besar
Q1	Membagi data menjadi empat bagian sama besar
Q2	Membagi data menjadi catur bagian sama besar

Dilakukan pembentukan N-Gram pada query kunci dan jawaban mahasiswa. N-gram merupakan teknik pemisahan teks menjadi *string* dengan panjang n mulai dari posisi tertentu dalam suatu teks. Pada penelitian ini bentuk *N-Gram* yang digunakan adalah Bigram (1 *String*) dan Unigram (2 *String*). Dengan adanya *N-Gram* ini diharapkan nantinya *system* tidak hanya mengoreksi persamaan setiap kata serta mengetahui kedekatan dari masing masing kata.

Tabel 2. Data Big Gram dan Unigram

<b>Bigram</b>	<b>Unigram</b>
Nilai	Nilai bagi
Bagi	Bagi data
Data	Data urut
Urut	Urut dalam
Dalam	Dalam empat
Empat	Empat bagi
Sama	Bagi sama
Besar	Sama besar
Jadi	Data jadi
Catur	Jadi empat
	Jadi catur
	Catur bagi

Selanjutnya dilakukan pembobotan kata menggunakan algoritma TF-IDF. TF (*Term Frequency*) merupakan frekuensi dari kemunculan term dalam dalam dokumen yang bersangkutan. IDF (*Inverse Document Frequensy*) digunakan untuk mengetahui bagaimana term tersebar secara luas pada dokumen jawaban.

Tabel 3. Data TF-IDF

Term	Q	D1	D2	df	D/df	IDF
nilai	1	0	0	1	3	1,477
bagi	2	2	2	6	0,5	0,698
data	1	1	1	3	1	1
urut	1	0	0	1	3	1,477
dalam	1	0	0	1	3	1,477
empat	1	1	0	2	1,5	1,176
sama	1	1	1	3	1	1
besar	1	1	1	3	1	1
jadi	0	1	1	2	1,5	1,176
catur	0	1	1	2	1,5	1,176
nilai bagi	1	0	0	1	3	1,477
bagi data	1	1	1	3	1	1
data urut	1	0	0	1	3	1,477
urut dalam	1	0	0	1	3	1,477
dalam empat	1	0	0	1	3	1,477
empat bagi	1	1	0	2	1,5	1,176
bagi sama	1	1	1		1	1
sama besar	1	1	1	3	1	1
data jadi	0	1	1	2	1,5	1,176
jadi empat	0	1	0	1	3	1,477
jadi catur	0	0	1	1	3	1,471
catur bagi	0	0	1	1	3	1,477

Untuk mencari nilai TF dilakukan dengan menghitung kemunculan term pada setiap dokumen. Untuk mencari nilai IDF dilakukan dengan menghitung  $\log_{10}$  dari rata-rata pada setiap term. Untuk mengetahui hasil TF-IDF dilakukan dengan mengalikan hasil IDF dengan term pada masing-masing dokumen.

Tabel 4. Hasil Perhitungan TF-IDF

Q	d1	d2
1,477	0	0
1,394	1,397	1,397
1	1	1
1,477	0	0
1,477	0	0
1,176	1,176	0
1	1	1
1	1	1
0	1,176	1,176
0	1,176	1,176
1,477	0	0
1	1	1
1,477	0	0
1,477	0	0
1,477	0	0
1,176	1,176	0
1	1	1
1	1	1
0	1,176	1,176
0	1,477	0
0	0	1,477
0	0	1,477

Setelah didapatkan data hasil Perhitungan TF-IDF selanjutnya adalah membentuk data D1 dan D2 dalam bentuk Matriks

Tabel 5. Perhitungan Matriks A

	0	0
	1,3979	1,3979
	1	1
	0	0
	0	0
	1,1761	0
	1	1
	1	1
	1,1761	1,1761
Matrik A =	0	1,4771
	0	0
	1	1
	0	0
	0	0
	0	0
	1,1761	0
	1	1
	1	1
	1,1761	1,1761
	1,4771	0
	0	1,4771
	0	1,4771

Setelah didapatkan nilai dari matriks A selanjutnya adalah melakukan perhitungan terhadap matriks eigenvector sehingga didapatkan hasil sebagai berikut :

Tabel 6. Nilai Matriks U

	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	-0,378	-0,031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,27	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	-0,153	-0,36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,27	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,27	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,318	-0,258	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,207	0,42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	-0,270	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,153	-0,36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,27	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,27	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,27	-0,021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,318	-0,258	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,192	-0,452	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,207	0,420	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-0,207	0,420	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Nilai dari matriks U akan dilakukan pengolahan sehingga didapatkan nilai dari masing-masing eigen baik eigenvector maupun eigen values Membentuk ruang vektor q yang dibuat dengan tahapan yang sama dalam membuat matrik A. Vektor query dibentuk ke dalam ruang berdimensi 2 dengan transformasi Proses ini diulang sebanyak jumlah sinonim yang terbentuk sehingga dibentuk hasil dari query seperti berikut.

Tabel 7. Tabel Perhitungan matriks U

$$Q = | -0,481 \ -0,425 |$$

$$D1 = | -0,679 \ -0,958 |$$

$$D2 = | -0,719 \ -0,401 |$$

Setelah hasil dari masing-masing query ditemukan dilanjutkan perhitungan *Cosine Similarity* dari matrik vector masing-masing D terhadap Q dengan perhitungan sebagai berikut :

$$D1 = \frac{(-0.481)(-0.679) + (-0.425)(-0.958)}{\sqrt{(-0.481)^2 + (-0.679)^2} \sqrt{(-0.425)^2 + (-0.958)^2}} = 0,9735$$

$$D2 = \frac{(-0.481)(-0.719) + (-0.425)(-0.401)}{\sqrt{(-0.481)^2 + (-0.719)^2} \sqrt{(-0.425)^2 + (-0.401)^2}} = 0,3319$$

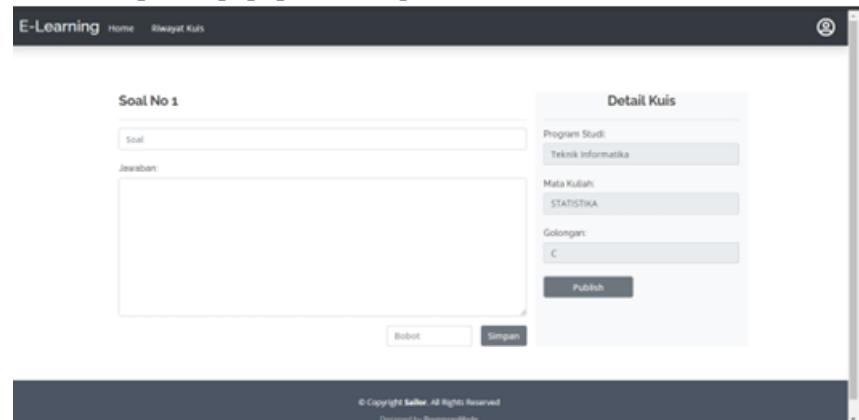
Dari kedua perhitungan sudut kemiripan tersebut dapat disimpulkan bahwa dokumen ke-1 memiliki nilai cosinus paling besar yang berarti memiliki nilai paling besar. Kemudian dari hasil cosinus tersebut dilakukan perhitungan nilai dengan mengalikan bobot dari setiap soal.

Dari hasil perhitungan tersebut selanjutnya adalah dilakukan implementasi dan pengujian terhadap aplikasi yang dikembangkan,



Gambar 2. Dashboard Aplikasi

Halaman ini merupakan halaman dashboard setelah login dengan hak akses admin. Pada halaman ini menampilkan materi yang telah ditambahkan dosen, kuis yang sedang berjalan. Terdapat “buat kuis” apabila dipilih akan menampilkan popup tambah quis.



Gambar 3. Menambahkan soal

Pada gambar 3 dosen dapat menambahkan soal beserta dengan jawaban yang diinginkan, soal tersebut akan muncul secara random pada lembar kerja mahasiswa, sedangkan jawaban yang diinputkan oleh dosen akan menjadi acuan bagi jawaban yang akan diisikan oleh mahasiswa.

Selanjutnya adalah melakukan proses pengukuran hasil pengujian pada aplikasi yang dikembangkan, pengujian dilakukan kepada 10 mahasiswa. Setelah system menghitung nilai untuk setiap mahasiswa dilakukan perbandingan nilai dengan nilai manual yang dilakukan dengan penilaian berdasarkan kriteria dosen dengan hasil sebagai berikut

Tabel 8. Hasil pengujian

No	Mahasiswa	Nilai Dosen				Nilai Sistem				Toal Selisih Nilai			
		Soal 1	Soal 2	Soal 3	TTL	Soal 1	Soal 2	Soal 3	TTL	Soal 1	Soal 2	Soal 3	TTL
1	Mahasiswa 1	33,3	33,3	33,3	100	33,3	33,3	33,3	100	0	0	0	0
2	Mahasiswa 2	20	5	20	45	0	0	33,3	33	20	5	13,3	38,3
3	Mahasiswa 3	33,3	33,3	33,3	100	33	33	33	99	0,3	0,3	0,3	0,9
4	Mahasiswa 4	33,3	33,3	33,3	100	33,3	33,3	28,69	95	0	0	4,61	4,61
5	Mahasiswa 5	33,3	33,3	33,3	100	33,15	33,3	33,3	100	0,15	0	0	0,15
6	Mahasiswa 6	33,3	33,3	33,3	100	33,3	33,3	33,3	100	0	0	0	0
7	Mahasiswa 7	15	15	33,3	63	33,3	33,3	33,3	100	18,3	18,3	0	36,6
8	Mahasiswa 8	33,3	15	15	63	33,3	33,3	33,3	100	0	18,3	18,3	36,6
9	Mahasiswa 9	20	10	20	50	32,5	0	32,2	65	12,5	10	12,2	34,7
10	Mahasiswa 10	20	20	20	60	33,2	0	0	33	13,2	20	20	53,2
Total Semua Selisih										64,45	71,9	68,71	205,06
Total Maksimal Selisih										333	333	333	1000
Persentase Selisih										19,30%	21,50%	20,60%	20,50%
Persentase Kesamaan										80,70%	78,50%	79,40%	79,50%

Berdasarkan hasil perbandingan pada table 8 nilai yang di proses oleh system dengan nilai dari dosen didapatkan pada soal 1 dengan persetanse selisih 19,30% dan persentase kesamaan 80.70%, pada soal 2 dengan persentase selisih 21.50% dan persentase kesamaan 78.50%, pada soal 3 dengan persentase selisih 20.60% dan persentase kesamaan 79.40%, pada total nilai mendapatkan persentase selisih 20.50% dan persentase kesamaan 79.50%.

Meskipun hasil pengujian menunjukkan adanya selisih rata-rata sebesar 20.5% antara skor sistem dan penilaian manual dosen, tingkat kesamaan sebesar 79.5% masih dapat dikategorikan cukup baik dalam konteks penilaian otomatis esai berbasis semantik. Hal ini dikarenakan penilaian esai secara intrinsik bersifat subjektif dan sangat dipengaruhi oleh persepsi penilai manusia, sehingga bahkan antar dosen sekalipun seringkali ditemukan perbedaan skor yang signifikan. Dalam penelitian-penelitian sebelumnya terkait penilaian otomatis esai, angka kesesuaian di atas 75% umumnya telah dianggap sebagai indikasi bahwa sistem mampu menangkap makna dan relevansi jawaban dengan cukup baik. Oleh karena itu, meskipun masih terdapat gap, tingkat akurasi yang diperoleh sistem ini dinilai sudah memadai untuk mendukung proses penilaian awal atau sebagai alat bantu dosen dalam mengevaluasi jawaban esai secara lebih efisien. Selanjutnya, pengembangan lebih lanjut dengan memperluas korpus sinonim dan memperbaiki model semantik diharapkan dapat meningkatkan akurasi sistem ke depan.

#### IV. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan terhadap Rancang Bangun Sistem Koreksi Otomatis yang mengintegrasikan algoritma *Generalized Latent Semantic Analysis* (GLSA) dan pendekatan Thesaurus pada dokumen esai, diperoleh sejumlah temuan signifikan. Diimplementasikan dalam bentuk platform e-learning berbasis web. Implementasi algoritma GLSA memungkinkan sistem untuk melakukan penilaian semantik terhadap jawaban mahasiswa secara otomatis, di mana durasi pemrosesan berkorelasi positif dengan kompleksitas struktur kalimat serta jumlah sinonim yang teridentifikasi. Berdasarkan uji coba terhadap 10 mahasiswa dan 3 soal, sistem menunjukkan tingkat kesesuaian penilaian dengan dosen sebesar 79.50%, dengan rata-rata deviasi sebesar 20.50%. Selanjutnya, integrasi algoritma Levenshtein Distance memungkinkan sistem mendeteksi kesalahan pengetikan dengan akurasi pada tingkat perbedaan satu karakter antar kata. Dalam kondisi tersebut, kata tetap diklasifikasikan sebagai valid, sehingga meningkatkan toleransi sistem terhadap variasi penulisan tanpa mengurangi akurasi penilaian secara keseluruhan.

## V. DAFTAR PUSTAKA

- [1] M. I. Thohir, A. P. Iskandar, I. L. Kharisma, and A. Fergina, "Implementasi gerakan wajah pada sistem ujian online menggunakan algoritma convolutional neural network," *J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 483–495, 2024, doi: 10.37859/coscitech.v5i2.7270.
- [2] R. Setiawan, "Optimasi Pengalaman Pengguna : Penilaian Otomatis dan Pencegahan Kecurangan Ujian Online," *Bit-Tech (Binary Digit. - Technol.)*, vol. 7, no. 2, pp. 299–306, 2024, doi: 10.32877/bt.v7i2.1758.
- [3] B. Putra, I. K. D. Nuryana, and R. A. J. Firdaus, "Rancang Bangun Aplikasi Koreksi Lembar Jawaban Komputer Menggunakan Metode Deteksi Tepi Canny," *INOVATE*, vol. 04, no. 01, pp. 16–24, 2019.
- [4] L. Hadibrata and T. H. Rochadiani, "Deteksi Potensi Menyontek Menggunakan Feedforward Neural Network Pada Ujian Daring," *SINTECH (Science Inf. Technol. J.)*, vol. 7, no. 2, pp. 92–100, 2024, doi: 10.31598/sintechjournal.v7i2.1585.
- [5] F. R. Awad Ahmed, T. E. Ahmed, R. A. Saeed, H. Alhumyani, S. Abdel-Khalek, and H. Abu-Zinadah, "Analysis and challenges of robust E-exams performance under COVID-19," *Results Phys.*, vol. 23, no. 1, pp. 1–7, 2021, doi: 10.1016/j.rinp.2021.103987.
- [6] I. Anas and S. Zakir, "Artificial Intelligence: Solusi Pembelajaran Era Digital 5.0," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 8, no. 1, pp. 35–46, 2024.
- [7] M. A. Bagaskoro and M. Mufti, "Penerapan Algoritma Levenshtein Distance, N-Gram dan Synonym Replacement Pada Sistem Penilaian Esai," in *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, 2024, vol. 3, no. 2, pp. 451–460.
- [8] P. Ghate, B. Jadhav, S. Kulkarni, P. Chopade, and P. Kota, "Global-Local Self-Attention Based Long Short-Term Memory With Optimization Algorithm for Speaker Identification," *IIUM Eng. J.*, vol. 26, no. 1, pp. 278–292, 2025, doi: 10.31436/IIUMEJ.V26I1.3386.
- [9] F. B. Wicaksono and Y. Yamasari, "Pengembangan Model Pengawas Berbasis Kecerdasan Buatan untuk Ujian Online," *J. Informatics Comput. Sci.*, vol. 06, no. 03, pp. 882–890, 2025.
- [10] R. F. Pasandaran, I. Suciat, and A. Alfisyahra, "Exploration of Students' Understanding of Geometry Concepts," *J. Penelit. Ilmu Pendidik.*, vol. 14, no. 2, pp. 148–157, 2021, doi: 10.21831/jpipfip.v14i2.39562.