BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 Penilaian Esai Otomatis (*Automated Essay Scoring*)

Salah satu solusi yang ditawarkan untuk masalah ini adalah *Automated Essay Scoring* (AES), yang didefinisikan sebagai tugas *machine learning* dalam proses pemrosesan bahasa natural, menciptakan suatu model yang dapat secara otomatis memberikan nilai pada jawaban mahasiswa pada esai (Rajagede, 2021). Rodriguez dkk dalam penelitian (Ayu Pradani dkk., 2023) AES berfungsi untuk memberikan skor pada esai berdasarkan fitur yang diekstrak dari teks esai. Proses penilaian terdiri dari dua tahap. Tahap pertama melibatkan pelatihan model untuk melakukan ekstraksi fitur yang akan digunakan untuk menilai esai. Tahap kedua melibatkan pembuatan model mesin AES yang dilatih di data berlabel berdasarkan fitur dari kumpulan data yang dipilih (Beseiso et al., 2020).

2.1.2 *Cosine Similarity*

Menurut Arfandy & Musdar dalam (Lahitani, 2022) *Cosine Similarity* adalah algoritma matematis canggih dalam bidang pemrosesan bahasa alami *(Natural Language Processing)* yang digunakan untuk menganalisis tingkat kemiripan antara dua dokumen teks. Dalam konteks evaluasi pendidikan, algoritma ini memiliki peran strategis untuk membandingkan kunci jawaban resmi dengan jawaban yang diberikan oleh mahasiswa secara otomatis dan objektif. Cara kerja *Cosine Similarity* didasarkan pada representasi vektor dari dokumen teks, di mana setiap dokumen ditransformasikan menjadi ruang vektor multidimensi. Algoritma

ini menghitung sudut antara dua vektor dokumen, dengan hasil perhitungan berkisar antara 0 (tidak mirip sama sekali) hingga 1 (identik). Semakin mendekati 1, menandakan tingkat kemiripan yang semakin tinggi (Sihombing, 2022). Perhitungan dalam *Cosine Similarity* ditunjukan pada persamaan (1) (Sihombing, 2022).

$$Similarity = Cos(0) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

Keterangan:

Cos(θ) : Cosinus sudut antara dua vektor.

A.B : Dot product (hasil kali titik) antara vektor A dan B.

||A|| : Magnitude (panjang) dari vektor A.

||B|| : Magnitude (panjang) dari vektor B.

Σ : Notasi sigma, menunjukkan penjumlahan.

n : Jumlah dimensi atau elemen dalam vector.

2.1.3 *Large Language Model* (LLM)

Large Language Model (LLM) adalah model kecerdasan buatan canggih yang dibangun menggunakan arsitektur *transformer deep learning*, mampu memproses dan menghasilkan bahasa natural dengan tingkat kompleksitas yang tinggi (Mishra dkk., 2024). *Model* (LLM) bukanlah pengganti penilaian manusia, melainkan mitra kolaboratif yang dapat memberikan penilaian objektif,

mengurangi beban administratif dosen, menyediakan perspektif tambahan dalam evaluasi. Penelitian ini membuka jendela baru dalam memahami potensi LLM sebagai alat evaluasi akademik yang canggih, sekaligus menekankan pentingnya pendekatan kolaboratif antara teknologi dan keahlian manusia (Ishida dkk., 2023).

2.1.4 *Text Mining*

Text Mining merupakan suatu cabang ilmu dalam *data mining* yang fokus pada analisis dokumen teks. Metode ini memungkinkan komputer untuk melakukan analisis teks secara otomatis guna mengekstraksi informasi berkualitas dari rangkaian teks yang terdapat dalam sebuah dokumen (Deolika & Taufiq Luthfi, 2019). Konsep dasar *Text Mining* adalah menemukan dan mengungkap pola-pola informasi yang tersembunyi dalam teks yang tidak terstruktur. Sebelum melakukan analisis lanjutan, diperlukan tahapan *text preprocessing* yang mencakup beberapa teknik penting seperti *tokenizing* (memecah teks menjadi unit-unit kata), *case folding* (mengubah teks menjadi huruf kecil), penghapusan *stopwords* (membuang kata-kata umum yang tidak memiliki makna signifikan), dan *stemming* (mengubah kata ke dalam bentuk dasarnya). Setelah proses *preprocessing* selesai, tahap selanjutnya adalah menggunakan metode klasifikasi untuk mengelompokkan teks ke dalam kategori-kategori tertentu sesuai dengan karakteristik dan pola yang ditemukan (Rio Feriangga Kurniawan, 2022).

2.1.5 *Text Preprocessing*

Text preprocessing adalah tahap awal pengolahan teks yang bertujuan mengubah data mentah menjadi format yang lebih terstruktur dan bermakna. Proses

ini melibatkan pemecahan teks menjadi unit-unit yang lebih kecil dan bermakna, dengan tingkat kompleksitas yang dapat disesuaikan sesuai kebutuhan analisis (Sihombing, 2022).

1. *Tokenization*

Tokenization adalah proses memecah kalimat atau string menjadi beberapa bagian (token) dengan menghapus tanda baca dan tanda hubung (Abiyyu Musa & Sibaroni, 2022). Menurut (Gupta, 2015), *tokenization* merupakan tugas memisahkan string teks penuh menjadi daftar kata-kata terpisah. Tahapan ini merupakan salah satu langkah kunci dalam preprocessing teks yang bertujuan untuk mempersiapkan data teks sebelum analisis lebih lanjut. Karakteristik *Tokenization*:

a. Memisahkan kalimat menjadi kata-kata atau term
b. Menghapus tanda baca seperti spasi, titik, koma, tanda tanya, dan tanda seru
c. Memudahkan proses analisis teks selanjutnya

2. *Lower Casing*

Proses mengubah seluruh huruf dokumen teks menjadi huruf kecil dikenal sebagai *lower casing*. Menurut (Slamet dkk., 2018) proses ini dimaksudkan untuk menghilangkan variasi kapitalisasi dan menyeragamkan format teks.

Beberapa manfaat utama dari *lower casing* dalam proses *text preprocessing* adalah sebagai berikut:

a. Menstandarisasi format teks
b. Mengurangi variasi yang tidak perlu
c. Mempermudah proses analisis selanjutnya.

3. *Stopword Removal*

Stopword removal adalah proses penghapusan kata-kata umum yang tidak memiliki makna signifikan dalam analisis teks. Menurut (Pradana & Hayaty, 2019), *stopwords* adalah kata-kata yang sering muncul dan tidak memberikan kontribusi makna penting dalam dokumen, seperti "yang", "di", "untuk", dan "dari". Proses ini bertujuan menyederhanakan teks dengan menghapus kata-kata yang tidak memberikan informasi bermakna. (Rosid dkk., 2020a) menjelaskan bahwa *library* Sastrawi menyediakan fungsionalitas *stopword removal* yang efisien untuk bahasa Indonesia. Proses *Stopword Removal* dengan *Library* Sastrawi:

a. Inisialisasi dilakukan dengan menggunakan *StopWordRemover* dari Sastrawi untuk membuat *instance* yang akan digunakan dalam penghapusan *stopwords*.
b. *Input* Teks disiapkan dengan menyediakan teks yang akan diproses.
c. Penghapusan Stopwords dilakukan dengan memanfaatkan metode *remove* untuk menghapus kata-kata yang ada dalam daftar *stopwords*.
d. *Output* dilakukan dengan menampilkan teks yang telah dibersihkan dari *stopwords*.

Beberapa penelitian mengungkapkan pentingnya *stopword removal*:

a. Mengurangi dimensi ruang fitur (Pradana & Hayaty, 2019)
b. Meningkatkan efisiensi komputasi (Zin dkk., 2017)
c. Membersihkan dokumen dari kata-kata tidak bermakna (Rosid dkk., 2020b)

4. *Lemmatization*

Lemmatization adalah proses dalam *text preprocessing* yang bertujuan untuk mengubah kata-kata yang terinfleksi menjadi bentuk dasarnya atau lemma. Proses

ini mempertimbangkan konteks dan makna kata, sehingga menghasilkan bentuk kata yang valid dalam bahasa. Berbeda dengan *stemming*, *lemmatization* memastikan hasil reduksi merupakan kata yang valid dan bermakna dalam bahasa. *Lemmatization* memainkan peran penting dalam meningkatkan relevansi dan akurasi hasil pencarian dalam pengumpulan informasi (IR). Menurut (Pramana dkk., 2022), *lemmatization* dan *stemming* adalah dua metode *preprocessing* yang digunakan untuk memastikan bahwa proses pencarian informasi tidak terganggu oleh variasi kata yang berbeda. Studi ini menunjukkan bahwa *lemmatization* biasanya lebih efektif dibandingkan dengan *stemming* dalam tugas-tugas yang berkaitan dengan kesamaan kalimat karena mempertimbangkan makna dan konteks kata.

2.1.6 *Term Frequency* (TF)

Term Frequency (TF) adalah ukuran statistik yang menunjukkan seberapa sering suatu istilah (kata) muncul dalam sebuah dokumen. Secara matematis, TF untuk suatu istilah *t* dalam dokumen *d* dapat dihitung dengan beberapa cara, namun yang paling umum adalah frekuensi mentah (*raw frequency*) atau frekuensi yang dinormalisasi. Frekuensi mentah adalah jumlah kemunculan istilah *t* dalam dokumen *d* . Sementara itu, frekuensi yang dinormalisasi bertujuan untuk mencegah bias terhadap dokumen yang lebih panjang, di mana suatu istilah mungkin muncul lebih sering hanya karena dokumen tersebut lebih Panjang (Amalia1 dkk., 2021)

Persamaan untuk menghitung *Term Frequency* (TF) disajikan dalam persamaan (2) (Amalia1 dkk., 2021).

$$W_{TF}(t_j, d_j) = f(t_j, d_j) \quad (2)$$

Keterangan:

$W_{TF}(t_j, d_j)$: Nilai TF dari term ke- pada dokumen ke-j

$f(t_j, d_j)$: Jumlah kemunculan dari term ke- pada dokumen ke-j

2.1.7 *Mean Absolute Percentage Error* (MAPE)

Mean Absolute Percentage Error (MAPE) adalah salah satu ukuran akurasi peramalan yang paling banyak digunakan karena keunggulannya dalam independensi skala dan kemudahan interpretasi. MAPE adalah metrik yang sering digunakan untuk mengukur seberapa jauh prediksi menyimpang dari nilai aktual, dalam bentuk persentase. Makin kecil nilai MAPE mendekati 0 maka makin akurat model prediksi. MAPE cocok untuk data dengan skala berbeda karena hasilnya bersifat proporsional. Persamaan untuk menghitung MAPE disajikan dalam persamaan (3) (Jusman dkk., 2022).

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100\% \quad (3)$$

Keterangan:

At : Nilai aktual periode ke-t

FtF_tFt : Nilai prediksi periode ke-t

nnn : Jumlah observasi

Menurut beberapa studi dan standar umum seperti yang terlihat dalam penelitian Indonesia, berikut kategori MAPE:

Tabel 2. 1 Kategori MAPE (Hidayanti dkk., 2022)

Rentang MAPE	**Kategori**
MAPE ≤ 10 %	Sangat akurat
10 % < MAPE ≤ 20 %	Baik
20 % < MAPE ≤ 50 %	Sedang
MAPE > 50 %	Buruk

Kemudian penelitian lainnya (Hidayanti dkk., 2022) menyatakan: “semakin kecil nilai kesalahan maka semakin akurat teknik peramalan. Tabel kategori: $x < 10\,\%$ = Sangat Baik; $10 \leq x < 20\,\%$ = Baik; $20 \leq x < 50\,\%$ = Cukup Baik; $x \geq 50\,\%$ = Buruk.

2.2 *State of the Art*

Penelitian-penelitian terkait penilaian esai otomatis telah berkembang pesat dalam beberapa tahun terakhir. Berbagai metode dan algoritma telah diterapkan untuk meningkatkan akurasi dan efisiensi proses evaluasi jawaban esai mahasiswa. *State of the Art* pada Tabel 2.2 yang disajikan menunjukkan beragam pendekatan yang telah dikaji untuk menghasilkan sistem penilaian esai otomatis yang lebih akurat dan handal.

Tabel 2. 2 State of the Art

No	**Penelitian**		**Hasil penelitian**			
1	**Peneliti (tahun)**	(Suryawan dkk., 2024)	**Akurasi**	**Hasil**	**Kelebihan**	**Kekurangan**
	Algoritma	*Stemming* Nazief-Adriani, Metode *Cosine Similarity*	94,3%	Berbasis android	Mengurangi subjektivitas	Membutuhkan kapasitas komputasi tinggi
2	**Peneliti (tahun)**	(Putra Pane dkk., 2024)	77,5%	Sistem aplikasi berbasis web	Metode inovatif	Terbatas pada *dataset* bahasa Indonesia
	Algoritma	*Text Mining, Cosine Similarity*				

No	Penelitian		Hasil penelitian			
3	**Peneliti (tahun)**	(Hidayatul Ula & Yuliyanti, 2024)	-	Aplikasi web "*Essay Checker*"	Mengurangi subjektivitas penilaian manual	Membutuhkan penyempurnaan algoritma penilaian teks
	Algoritma	*Cosine Similarity*				
4	**Peneliti (tahun)**	(Yulita dkk., 2023)	0.132 (MAE)	Sistem penilaian otomatis	Mengembangkan metode TF-IDF	Sistem masih mengalami kesulitan membedakan jawaban yang memiliki makna sama namun menggunakan kata-kata berbeda
	Algoritma	*Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity*				
5	**Peneliti (tahun)**	(Lahitani, 2022)	52%	Sistem penilaian esai otomatis berbasis teks	Penilaian cepat dan objektif menggunakan *text mining.*	Keterbatasan dalam menghasilkan skor proporsional
	Algoritma	*Cosine Similarity*				
6	**Peneliti (tahun)**	(Sihombing, 2022)	90,58%	Sistem penilaian esai otomatis		Keterbatasan dalam memahami nuansa

No	Penelitian		Hasil penelitian			
	Algoritma	*Cosine Similarity*		berbasis nlp	Hasil yang lebih cepat, objektif, dan konsisten	semantik kompleks dan variasi jawaban yang sangat beragam
7	**Peneliti (tahun)**	(Rosnelly dkk., 2021)	89,5%	Sistem penilaian esai otomatis berbasis *text mining*	Hasil objektif, dan konsisten	Keterbatasan dalam memahami sinonim dan variasi kata
	Algoritma	*Cosine Similarity* dan Nazief-Adriani *Algorithms*				
8	**Peneliti (tahun)**	(Mufiid dkk., 2021)	8.94 (RMSE)	Aplikasi berbasis web	Mengatasi variasi jawaban siswa	Keterbatasan akurasi akibat kendala seperti perbedaan similaritas jawaban, kesalahan pengetikan, dan hilangnya beberapa kata pada proses filtering stopword.
	Algoritma	*Cosine Similarity* dan *Synonym Recognition*				

Secara keseluruhan, penelitian-penelitian yang dipaparkan pada Tabel 2.2 dan penelitian-penelitian lain yang digunakan sebagai referensi penelitian ini telah dilakukan sitasi pada latar belakang menunjukkan pemanfaatan metode *Cosine Similarity* yang efektif dalam pengembangan model penilaian esai otomatis. Secara umum, penelitian-penelitian tersebut menunjukkan bahwa: (1) *Cosine Similarity* banyak digunakan sebagai metode utama penilaian kemiripan jawaban mahasiswa dan kunci jawaban, dengan akurasi berkisar 52%-94,3%. (2) Beberapa penelitian mengombinasikan teknik *preprocessing* seperti *stemming* Nazief-Adriani, TF-IDF, atau *synonym recognition* untuk meningkatkan akurasi. (3) Keterbatasan utama yang masih muncul Adalah sensitivitas terhadap sinonim dan variasi struktur kalimat, ketergantungan pada pola tekstual statis, belum adanya pemanfaatan analisis semantik mendalam.

Penelitian ini memposisikan diri dengan menambahkan lapisan analisis semantik menggunakan LLM, sehingga tidak hanya mengandalkan kesamaan leksikal, tetapi juga konteks dan kualitas argumentasi.

2.3 Matriks Penelitian

Matriks penelitian pada Tabel 2.3 menunjukkan berbagai penelitian yang telah dilakukan, dengan informasi tentang penulisnya, tahun publikasi, serta algoritma yang digunakan. Penelitian mencakup berbagai aspek tujuan penelitian seperti pengembangan model, perhitungan, dan perbandingan algoritma.

Tabel 2. 3 Matriks Penelitian

No	Penelitian	Tujuan Penelitian			Algoritma				
		Pengembangan Model	Perhitungan Kesamaan Jawaban	Perbandingan Algoritma	Nazief-Adriani	*Cosine Similarity*	*Text Mining*	TF-IDF	*Synonym Recognition*
1	(Suryawan dkk., 2024)		√	√	√	√			
2	(Putra Pane dkk., 2024)	√	√			√	√		
3	(Hidayatul Ula & Yuliyanti, 2024)	√	√			√			
4	(Yulita dkk., 2023)		√			√		√	
5	(Lahitani, 2022)		√			√			
6	(Sihombing, 2022)		√			√			
7	(Rosnelly dkk., 2021)		√		√	√			
8	(Mufiid dkk., 2021)	√	√	√		√			√
9	*Our Research*	√	√			√			

Matriks penelitian pada Tabel 2.3 menunjukkan bahwa penelitian sebelumnya memiliki keterbatasan signifikan diantaranya sistem berbasis *Cosine Similarity* seperti pada (Rosnelly dkk., 2021) dan (Lahitani, 2022) gagal menangkap makna semantik mendalam, kemudian pemrosesan teks sederhana oleh (Ahmad et al., 2020) dan (Sihombing, 2022) terbatas pada pencocokan kata kunci tanpa mempertimbangkan koherensi argumentasi, dan

tidak adanya umpan balik kualitatif pada sistem (Putra Pane dkk., 2024) dan (Yulita dkk., 2023). Penelitian ini mengusulkan kombinasi *Cosine Similarity* dan *Model* (LLM) dan sistem pembobotan adaptif untuk mengatasi *gap* tersebut, sehingga dapat menghasilkan penilaian esai yang lebih akurat, kontekstual, dan disertai umpan balik konstruktif.