BAB II
KAJIAN LITERATUR

2.1 Penilaian Esai

Penilaian esai merupakan proses evaluasi terhadap jawaban atau karangan yang ditulis oleh peserta didik dalam bentuk teks. Berbeda dengan soal pilihan ganda yang memiliki jawaban tunggal, esai memberikan kebebasan kepada penulis untuk mengungkapkan pemikiran, argumen, dan pemahaman secara terstruktur. Penilaian esai bertujuan untuk mengukur kemampuan berpikir kritis, kemampuan komunikasi tertulis, serta kedalaman pemahaman materi yang dimiliki oleh peserta didik [20], [21].

2.1.1 Definisi Penilaian Esai

Penilaian esai didefinisikan sebagai proses pemberian skor atau penilaian kualitatif terhadap teks jawaban yang ditulis secara bebas oleh peserta didik berdasarkan kriteria tertentu yang telah ditetapkan sebelumnya. Menurut [21]. penilaian esai mencakup aspek konten, tata bahasa, koherensi, serta relevansi jawaban terhadap pertanyaan yang diajukan. Penilaian ini umumnya dilakukan oleh penilai manusia yang menggunakan rubrik sebagai acuan evaluasi, namun pendekatan tersebut memiliki keterbatasan berupa potensi subjektivitas dan inkonsistensi antar penilai [21].

Secara umum, penilaian esai dapat dikategorikan menjadi dua bentuk utama, yaitu penilaian holistik dan penilaian analitik. Penilaian holistik memberikan satu skor tunggal yang mencerminkan keseluruhan kualitas tulisan, sedangkan penilaian analitik mengevaluasi beberapa aspek secara terpisah seperti isi, organisasi, kosakata, penggunaan bahasa, dan mekanik penulisan. Kedua pendekatan ini memiliki kelebihan dan kekurangan masing-masing, dan pemilihannya bergantung pada tujuan evaluasi yang ingin dicapai [21].

2.1.2 Jenis Penilaian Esai

Berdasarkan cara pelaksanaannya, penilaian esai dapat dibedakan menjadi dua jenis utama. Pertama, penilaian esai manual (*human scoring*) yang dilakukan oleh penilai manusia secara langsung. Penilai membaca dan mengevaluasi jawaban esai berdasarkan kriteria atau rubrik yang telah ditentukan. Pendekatan ini dianggap paling akurat dalam menilai kualitas teks secara mendalam, namun memiliki kelemahan berupa waktu yang panjang, biaya tinggi, serta potensi subjektivitas antar penilai [20], [21].

Kedua, penilaian esai otomatis (*Automatic Essay Scoring/AES*) yang memanfaatkan teknologi komputer dan kecerdasan buatan untuk menilai jawaban esai secara otomatis. *AES*

dirancang untuk menggantikan atau melengkapi penilaian manual dengan menggunakan teknik pemrosesan bahasa alami *(NLP)* dan pembelajaran mesin untuk menganalisis dan mengevaluasi teks secara cepat, konsisten, dan terukur [21].

Selain berdasarkan cara pelaksanaannya, penilaian esai juga dapat dibedakan berdasarkan bentuk jawaban yang dinilai, yaitu esai panjang (*extended response*) yang memerlukan jawaban komprehensif dalam beberapa paragraf, dan esai pendek (*short answer*) yang hanya menuntut jawaban singkat namun tetap terstruktur [20], [21].

2.1.3 Tantangan dalam Penilaian Esai

Penilaian esai menghadapi berbagai tantangan yang kompleks, baik dalam pendekatan manual maupun otomatis. Pada penilaian manual, tantangan utama meliputi inkonsistensi antar penilai, di mana dua penilai yang berbeda dapat memberikan skor yang berbeda untuk esai yang sama. Selain itu, penilaian manual sangat memakan waktu dan sumber daya, khususnya ketika harus menilai ratusan atau ribuan jawaban secara bersamaan [21].

Pada penilaian esai otomatis, tantangan yang dihadapi meliputi kemampuan sistem untuk memahami makna teks secara mendalam, bukan sekadar mencocokkan kata kunci. Sistem *AES* harus mampu menangani variasi gaya penulisan, sinonim, parafrase, serta struktur kalimat yang beragam. Tantangan lainnya mencakup keterbatasan data pelatihan yang berlabel, terutama untuk bahasa Indonesia yang masih relatif sedikit memiliki dataset penilaian esai yang tersedia secara public [21], [22].

Tantangan spesifik lainnya dalam AES berbahasa Indonesia meliputi kompleksitas morfologi bahasa Indonesia, penggunaan kata serapan dari bahasa daerah dan bahasa asing, serta variasi ejaan yang tidak standar. Hal ini menjadikan pengembangan sistem *AES* untuk Bahasa Indonesia memerlukan pendekatan yang lebih adaptif dan model bahasa yang telah dioptimalkan untuk karakteristik bahasa tersebut [21].

2.1.4 Contoh Esai Yang Dipakai

Pada penelitian ini, jenis esai yang digunakan sebagai objek penilaian adalah esai pendek (*short essay*) atau dikenal juga sebagai esai uraian terbatas (*restricted response essay*). Esai pendek merupakan jenis soal uraian yang membatasi panjang dan ruang lingkup jawaban, umumnya hanya terdiri dari satu hingga beberapa kalimat [11]. Jenis esai ini bertujuan untuk mengukur pemahaman mahasiswa terhadap suatu konsep secara ringkas dan langsung pada inti jawaban, sehingga memiliki jawaban referensi yang relatif baku dan cocok untuk dinilai secara otomatis menggunakan pendekatan kemiripan semantic. [11], [22]

2.2 Automatic Essay Scoring

Automatic Essay Scoring (AES) merupakan salah satu penerapan teknologi kecerdasan buatan dalam bidang pendidikan yang bertujuan untuk mengotomatisasi proses penilaian jawaban esai. Sistem ini dikembangkan sebagai solusi atas keterbatasan penilaian manual yang memerlukan waktu yang lama serta berpotensi menimbulkan subjektivitas antar penilai. Dengan teknik komputasi, *AES* mampu melakukan evaluasi terhadap teks secara lebih cepat, konsisten, dan terukur sehingga mendukung proses pembelajaran berbasis digital yang semakin berkembang [20], [21].

AES memiliki keterkaitan yang erat dengan bidang *Natural Language Processing* (*NLP*), karena proses penilaian esai melibatkan analisis teks secara otomatis. *NLP* digunakan untuk memahami struktur bahasa, tata bahasa, serta makna dari jawaban yang diberikan oleh peserta didik. Dengan teknik *machine learning* dan *deep learning*, sistem *AES* tidak hanya mampu mengevaluasi aspek permukaan teks, tetapi juga dapat memahami hubungan semantik antar kalimat sehingga menghasilkan penilaian yang lebih akurat dan relevan [23], [24]. *AES* dapat dibedakan menjadi dua jenis utama berdasarkan bentuk input yang diproses.

1. *AES* berbasis teks (*text-based AES*), yaitu sistem yang mengevaluasi jawaban esai dalam bentuk teks digital yang diketik oleh pengguna. Jenis ini merupakan pendekatan yang paling umum digunakan karena lebih mudah diproses secara langsung menggunakan teknik *NLP* [21], [22].
2. *AES* berbasis tulisan tangan (*handwriting-based AES*), yaitu sistem yang memanfaatkan teknologi *Optical Character Recognition* (*OCR*) untuk mengubah tulisan tangan menjadi teks digital sebelum dilakukan proses penilaian. Pendekatan ini memungkinkan sistem untuk digunakan pada ujian berbasis kertas yang kemudian didigitalisasi [19], [21], [25].

Perkembangan *AES* mengalami evolusi yang signifikan seiring kemajuan teknologi. pada tahap awal, sistem *AES* menggunakan pendekatan berbasis aturan (*rule-based*) dengan memanfaatkan fitur sederhana seperti panjang teks, jumlah kata, dan frekuensi istilah tertentu., pendekatan ini memiliki keterbatasan dalam memahami makna teks secara mendalam. Selanjutnya, pendekatan *machine learning* mulai digunakan untuk mempelajari pola dari data esai dan meningkatkan akurasi penilaian, meskipun masih bergantung pada ekstraksi fitur secara manual. Perkembangan terbaru didominasi oleh penggunaan *deep learning* berbasis arsitektur transformer, seperti *BERT* dan model turunannya, yang mampu memahami konteks bahasa secara *bidirectional* dan meningkatkan kualitas evaluasi esai secara signifikan [22], [26].

Pendekatan berbasis *Large Language Model* (*LLM*) juga mulai dimanfaatkan dalam sistem *AES* modern untuk meningkatkan akurasi serta memberikan umpan balik yang lebih informatif kepada pengguna. Hal ini menjadikan *AES* tidak hanya sebagai alat penilaian otomatis, tetapi juga sebagai sistem yang mampu mendukung proses pembelajaran secara adaptif dan interaktif [27], [28].

Agar suatu sistem dapat dikategorikan sebagai *Automatic Essay Scoring* yang baik, sistem tersebut harus memiliki beberapa karakteristik utama, yaitu mampu melakukan penilaian secara otomatis, memiliki representasi teks yang jelas, menggunakan metode evaluasi yang terukur dan menghasilkan skor yang dapat diinterpretasikan. sistem juga harus diuji menggunakan metrik evaluasi untuk memastikan kualitas hasil penilaian yang dihasilkan [21], [22].

2.3 Artificial Intelligence

Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang ilmu komputer yang berfokus pada pengembangan sistem dan mesin yang mampu melakukan tugas-tugas yang secara normal memerlukan kecerdasan manusia. AI mencakup kemampuan seperti penalaran, pembelajaran, pemecahan masalah, persepsi bahasa, dan pengambilan keputusan. Konsep AI pertama kali diperkenalkan oleh John McCarthy pada tahun 1956 dan sejak saat itu berkembang pesat menjadi salah satu bidang teknologi yang paling berpengaruh di era modern [23], [24].

2.3.1 Definisi Artificial Intelligence

Artificial Intelligence didefinisikan sebagai kemampuan mesin atau sistem komputer untuk meniru fungsi kognitif yang diasosiasikan dengan pikiran manusia, seperti belajar dari pengalaman, memahami bahasa alami, mengenali pola, dan membuat keputusan. Menurut Khurana et al. (2023), AI merupakan teknologi yang memungkinkan komputer untuk memahami, mengolah, dan menghasilkan informasi secara cerdas melalui berbagai teknik seperti machine learning, *deep learning*, dan pemrosesan bahasa alami [24].

Dalam perkembangannya, AI dapat dibagi menjadi tiga kategori utama berdasarkan kemampuannya. Pertama, *Narrow AI* (AI Sempit) yang dirancang untuk melakukan tugas spesifik tertentu dengan sangat baik, seperti sistem rekomendasi, pengenalan wajah, dan penerjemah mesin. Kedua, *General AI* (AI Umum) yang secara teoritis mampu melakukan berbagai tugas intelektual seperti manusia. Ketiga, Super *AI* yang merupakan konsep hipotetis tentang *AI* yang melampaui kecerdasan manusia dalam semua aspek [24].

2.3.2 Contoh Penerapan Artificial Intelligence

Artificial Intelligence telah diterapkan secara luas di berbagai bidang kehidupan. Dalam bidang kesehatan, AI digunakan untuk diagnosis penyakit melalui analisis citra medis seperti MRI dan *CT-scan*, serta untuk prediksi risiko penyakit berdasarkan data pasien. Dalam bidang transportasi, AI digunakan dalam pengembangan kendaraan otonom yang mampu berkendara tanpa pengemudi manusia. Dalam bidang keuangan, AI digunakan untuk deteksi penipuan (*fraud detection*), dan perdagangan algoritmik [24].

Dalam bidang pendidikan, AI telah memberikan dampak signifikan melalui berbagai aplikasi inovatif. Sistem tutoring cerdas (*Intelligent Tutoring System*) menggunakan *AI* untuk memberikan bimbingan belajar yang dipersonalisasi sesuai kebutuhan masing-masing siswa. *Chatbot* berbasis *AI* digunakan sebagai asisten belajar yang dapat menjawab pertanyaan mahasiswa kapan saja [29], [30]. Salah satu penerapan *AI* yang paling relevan dengan penelitian ini adalah sistem *Automatic Essay Scoring* (*AES*), yaitu sistem yang menggunakan AI untuk menilai jawaban esai secara otomatis sebagai alternatif dari penilaian manual yang membutuhkan waktu dan sumber daya besar [21], [24], [28].

Teknik-teknik utama yang digunakan dalam AI modern meliputi *machine learning*, *deep learning*, dan *Natural Language Processing* (*NLP*). *Machine learning* memungkinkan sistem untuk belajar dari data tanpa diprogram secara eksplisit. *Deep learning* menggunakan jaringan saraf tiruan berlapis (*neural networks*) untuk memproses data kompleks. *NLP* memungkinkan komputer untuk memahami dan menghasilkan bahasa manusia secara komputasional. Ketiga teknik ini saling melengkapi dan menjadi fondasi dari sistem *AES* modern yang dikembangkan dalam penelitian ini [23], [24].

2.4 Natural Language Processing

Natural Language Processing (*NLP*) merupakan cabang dari kecerdasan buatan yang berfokus pada kemampuan komputer untuk memahami, mengolah, dan menghasilkan bahasa manusia secara komputasional. *NLP* menggabungkan ilmu linguistik dan teknik pembelajaran mesin untuk memungkinkan sistem memproses teks atau ucapan secara otomatis sehingga dapat digunakan dalam berbagai aplikasi berbasis Bahasa [23], [24]. Dalam perkembangannya, *NLP* menjadi salah satu teknologi kunci dalam pengolahan data teks karena mampu menangani bahasa alami yang kompleks dan tidak terstruktur.

Pemrosesan bahasa alami dilakukan melalui beberapa tingkatan linguistik yang saling berkaitan. Pada tingkat paling dasar, *NLP* memproses teks melalui analisis morfologi yang berfokus pada pembentukan kata dari unit terkecil yang bermakna. Selanjutnya,

analisis sintaksis digunakan untuk memahami struktur kalimat berdasarkan aturan tata bahasa. Setelah itu, analisis semantik berperan dalam memahami makna dari kata dan kalimat, sementara analisis pragmatik mempertimbangkan konteks penggunaan bahasa untuk memperoleh interpretasi makna yang lebih tepat. Setiap tingkatan ini berkontribusi dalam membangun pemahaman bahasa yang lebih komprehensif dalam sistem *NLP* [23], [24], [31].

Seiring perkembangan teknologi, pendekatan *NLP* mengalami evolusi yang signifikan. Pada awalnya, *NLP* menggunakan metode berbasis aturan (*rule-based*) yang bergantung pada kaidah linguistik yang dirancang secara manual. Pendekatan ini kemudian berkembang menjadi metode statistik yang memanfaatkan probabilitas dalam memproses bahasa. Saat ini, *NLP* didominasi oleh pendekatan berbasis *deep learning* yang menggunakan model jaringan saraf untuk mempelajari representasi bahasa secara otomatis dari data dalam jumlah besar [23], [32], [33], [34].

Perkembangan terbaru dalam *NLP* ditandai dengan penggunaan arsitektur Transformer yang mampu menangkap hubungan konteks dalam teks secara lebih efektif melalui mekanisme *self-attention*. Model pra-latih seperti *BERT* dan variannya memungkinkan sistem memahami hubungan antar kata dalam kalimat secara *bidirectional*, sehingga meningkatkan performa pada berbagai tugas pemrosesan bahasa alami. Pendekatan ini memungkinkan *NLP* tidak hanya memahami teks berdasarkan kemunculan kata, tetapi juga berdasarkan makna yang terkandung secara lebih mendalam [23], [24], [33].

Dengan kemampuannya dalam memahami bahasa manusia secara komprehensif, *NLP* menjadi fondasi utama dalam berbagai sistem cerdas modern yang berbasis teks. Teknologi ini terus berkembang seiring meningkatnya kebutuhan akan sistem yang mampu memproses dan memahami informasi dalam bentuk bahasa alami secara akurat dan efisien [23], [32].

2.5 Metode Pengembangan Perangkat Lunak Waterfall

Metode *Waterfall* merupakan salah satu model dalam *Software Development Life Cycle* (SDLC) yang menggunakan pendekatan sistematis dan berurutan (sekuensial), di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini pertama kali diperkenalkan oleh Winston W. Royce dan masih banyak digunakan dalam pengembangan perangkat lunak yang memiliki kebutuhan sistem yang jelas dan stabil sejak awal [35].

Dalam metode *Waterfall*, proses pengembangan perangkat lunak terdiri dari beberapa tahapan yang dilakukan secara berurutan, yaitu analisis kebutuhan (*requirement analysis*), perancangan sistem (*system design*), implementasi (*coding*), pengujian (*testing*), dan pemeliharaan (*maintenance*). Tahap analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan pengguna secara menyeluruh. Selanjutnya, tahap perancangan sistem bertujuan untuk merancang arsitektur sistem, basis data, serta antarmuka pengguna. Tahap implementasi merupakan proses penerjemahan desain ke dalam kode program. Setelah itu dilakukan tahap pengujian untuk memastikan sistem berjalan sesuai dengan kebutuhan dan bebas dari kesalahan. Tahap terakhir adalah pemeliharaan, yaitu proses perbaikan dan pengembangan sistem setelah digunakan [36].

Metode *Waterfall* memiliki beberapa kelebihan, antara lain alur pengembangan yang jelas dan terstruktur, dokumentasi yang lengkap, serta mudah dipahami dan diterapkan. Selain itu, metode ini cocok digunakan pada proyek dengan kebutuhan yang telah terdefinisi dengan baik sejak awal, sehingga memudahkan pengelolaan dan pengendalian proyek [36].

Metode *Waterfall* juga memiliki beberapa kekurangan, seperti kurang fleksibel terhadap perubahan kebutuhan, sulit melakukan revisi pada tahap yang telah dilewati, serta risiko kesalahan yang tinggi apabila terjadi kesalahan pada tahap awal. Selain itu, hasil sistem baru dapat dilihat secara keseluruhan pada tahap akhir, sehingga kurang cocok untuk pengembangan sistem yang bersifat dinamis [36].

2.6 *BERT*

BERT (*Bidirectional Encoder Representations from Transformers*) merupakan model berbasis arsitektur *Transformer* yang dikembangkan oleh *Google* pada tahun 2018. *BERT* dirancang untuk memahami konteks bahasa secara *bidirectional*, yaitu dengan mempertimbangkan kata-kata di sebelah kiri maupun kanan dari setiap token secara bersamaan dalam satu urutan teks. Kemampuan ini menjadikan *BERT* unggul dibandingkan model-model sebelumnya dalam berbagai tugas *Natural Language Processing* (*NLP*) modern [23], [24].

Arsitektur Transformer dirancang untuk mengatasi keterbatasan model sekuensial seperti RNN dan LSTM. Transformer menggunakan mekanisme *self-attention* yang memungkinkan setiap token memperhatikan seluruh token lain dalam satu waktu. Struktur utamanya terdiri dari *encoder* dan *decoder*, dengan komponen utama berupa *Multi-Head*

Self-attention, *feed-forward network* dan *positional* en*coding* untuk mempertahankan informasi urutan kata [24], [31], [37].

Mekanisme inti *Transformer* dirumuskan pada persamaan

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \text{.....(1)}$$

Keterangan:

1. Q= Query
2. K= Key
3. V= Value
4. QK^T= perkalian matriks antara Qdan transpose dari K
5. K^T= transpose dari matriks K
6. d_k= dimensi dari vektor key

Q (Query), K (Key), dan V (Value) adalah representasi vektor, dan d_k adalah dimensi vektor untuk stabilisasi perhitungan [37].

BERT (*Bidirectional Encoder Representations from Transformers*) merupakan model berbasis *encoder Transformer* yang mampu memahami konteks secara dua arah. Setiap token direpresentasikan melalui kombinasi *token embedding*, *segment embedding*, dan *positional embedding* sebelum diproses dalam beberapa lapisan *encoder* untuk menghasilkan representasi kontekstual [24], [38], [39].

Seiring perkembangannya, *BERT* memiliki berbagai varian yang dirancang untuk meningkatkan efisiensi maupun performa. Berikut merupakan beberapa varian utama *BERT*:

RoBERTa (Robustly Optimized BERT Pretraining Approach) dikembangkan untuk mengatasi keterbatasan optimasi pada pelatihan BERT asli. Varian ini menghapus mekanisme *Next Sentence Prediction* (NSP) yang terbukti tidak terlalu berdampak signifikan terhadap performa akhir model, serta melakukan pelatihan dengan ukuran *batch* yang jauh lebih besar dan dataset yang lebih ekstensif. Dari segi arsitektur, RoBERTa pada dasarnya sama dengan BERT, namun modifikasi pada strategi pelatihannya dengan menerapkan *dynamic masking* membuatnya mampu mencapai performa yang lebih tinggi dibandingkan BERT standar dalam berbagai tugas *Natural Language Processing* [22].

DistilBERT diciptakan sebagai solusi untuk mengatasi masalah ukuran model BERT yang terlalu besar dan berat untuk dijalankan pada perangkat dengan sumber daya terbatas. Varian ini menggunakan teknik *knowledge distillation*, di mana model yang lebih kecil dilatih untuk meniru perilaku dan keluaran dari model BERT utuh. Arsitektur DistilBERT mengurangi jumlah lapisan (*layer*) transformer sebanyak 40% (dari 12 menjadi 6 lapisan) dan menghapus komponen *token-type embeddings* serta *pooler*. Keunggulan utamanya

adalah DistilBERT berukuran 40% lebih kecil, 60% lebih cepat dalam proses inferensi, namun masih mampu mempertahankan sekitar 97% kemampuan pemahaman bahasa dari model BERT asli [22].

ALBERT (A Lite BERT) dirancang untuk menjawab masalah konsumsi memori dan peningkatan parameter yang sangat masif pada BERT standar. ALBERT memodifikasi arsitektur dengan dua teknik utama: *factorized embedding parameterization* yang memisahkan ukuran *vocabulary embedding* dari *hidden layer*, serta *cross-layer parameter sharing* di mana seluruh lapisan transformer berbagi parameter bobot yang sama. Selain itu, ALBERT mengganti tugas NSP dengan *Sentence Order Prediction* (SOP). Perubahan arsitektur ini membuat ALBERT memiliki jumlah parameter yang jauh lebih sedikit dibandingkan BERT, sehingga lebih efisien secara komputasi tanpa mengorbankan akurasi yang signifikan [22].

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) hadir dengan pendekatan pelatihan yang berbeda dari BERT standar. Alih-alih menggunakan *Masked Language Modeling* (MLM) yang menutupi 15% kata, ELECTRA menggunakan pendekatan *Replaced Token Detection* (RTD). Arsitekturnya terdiri dari dua jaringan: generator yang menggantikan beberapa kata pada teks asli dengan kata pilihan model, dan diskriminator yang bertugas menebak apakah kata tersebut adalah kata asli atau hasil penggantian dari generator. Keunggulan dari pendekatan ini adalah model belajar dari seluruh token input, menjadikannya jauh lebih efisien dan mampu mencapai akurasi tinggi dengan data dan komputasi yang lebih sedikit dibandingkan BERT [22].

IndoBERT merupakan varian BERT yang secara khusus diadaptasi dan dilatih ulang menggunakan korpus bahasa Indonesia. BERT standar dilatih menggunakan data berbahasa Inggris, sehingga kurang optimal dalam memahami struktur, tata bahasa, dan konteks lokal bahasa Indonesia. Arsitektur IndoBERT pada dasarnya sama dengan BERT, namun proses *pre-training* dilakukan menggunakan dataset besar berbahasa Indonesia, seperti Indonesian Wikipedia dan teks berita (*Indo4B*). Keunggulan utama IndoBERT terletak pada kemampuannya yang jauh lebih baik dalam menangkap makna semantik dan konteks kata dalam bahasa Indonesia [22], [40].

Dengan berbagai varian tersebut, *BERT* menjadi arsitektur yang fleksibel dan banyak digunakan dalam berbagai tugas *NLP* modern seperti klasifikasi teks, *semantic similarity*, dan *Automatic Essay Scoring* [26], [38], [39].

2.6.1 Sentence-*BERT* (S*BERT*)

Sentence-*BERT* (S*BERT*) merupakan pengembangan dari model *BERT* yang dirancang khusus untuk menghasilkan representasi *embedding* pada tingkat kalimat yang bersifat semantik dan efisien untuk perhitungan kemiripan teks. Berbeda dengan *BERT* konvensional yang memerlukan pemrosesan pasangan kalimat secara bersamaan, S*BERT* memungkinkan setiap kalimat diproses secara independen sehingga *embedding* dapat disimpan dan digunakan kembali. Pendekatan ini secara signifikan meningkatkan efisiensi komputasi, terutama pada skenario perbandingan teks dalam jumlah besar seperti pada sistem *Automatic Essay Scoring* [41], [42].

S*BERT* mengubah teks menjadi vektor dalam ruang *embedding* berdimensi tetap, sehingga hubungan semantik antar kalimat dapat dihitung menggunakan metrik matematis seperti *cosine similarity* [41], [42]. S*BERT* tidak hanya mempertimbangkan kesamaan kata, tetapi juga kesamaan makna secara kontekstual [41], [43].

2.6.2 Konsep Dasar Sentence-*BERT*

S*BERT* dibangun dengan memodifikasi arsitektur *BERT* menggunakan pendekatan *siamese network* dan *triplet network*. Dalam pendekatan ini, dua atau tiga kalimat diproses melalui model *BERT* yang memiliki bobot yang sama (*shared weights*), sehingga menghasilkan *embedding* yang dapat dibandingkan secara langsung dalam ruang vektor yang sama [41]. Jika suatu kalimat dinotasikan sebagai S, maka proses *encoding SBERT* dapat dinyatakan sebagai:

$$u = SBERT(S) \quad \text{.....(2)}$$

Keterangan:

1. u= vektor hasil
2. S= teks input
3. SBERT(S)= proses *encoding* kalimat menggunakan model *Sentence-BERT*

di mana $u \in \mathbb{R}^d$ merupakan vektor *embedding* berdimensi tetap. Representasi ini dirancang agar kalimat dengan makna yang mirip memiliki jarak yang dekat dalam ruang vektor.

Untuk mengukur kesamaan antara dua kalimat S_1 dan S_2, digunakan *cosine similarity*:

$$\text{similarity}(u, v) = \frac{u \cdot v}{\|u\| \times \|v\|} \quad \text{.....(3)}$$

Keterangan:

1. u, v= dua vektor
2. similarity(u, v)= nilai kemiripan antara vektor u dan v

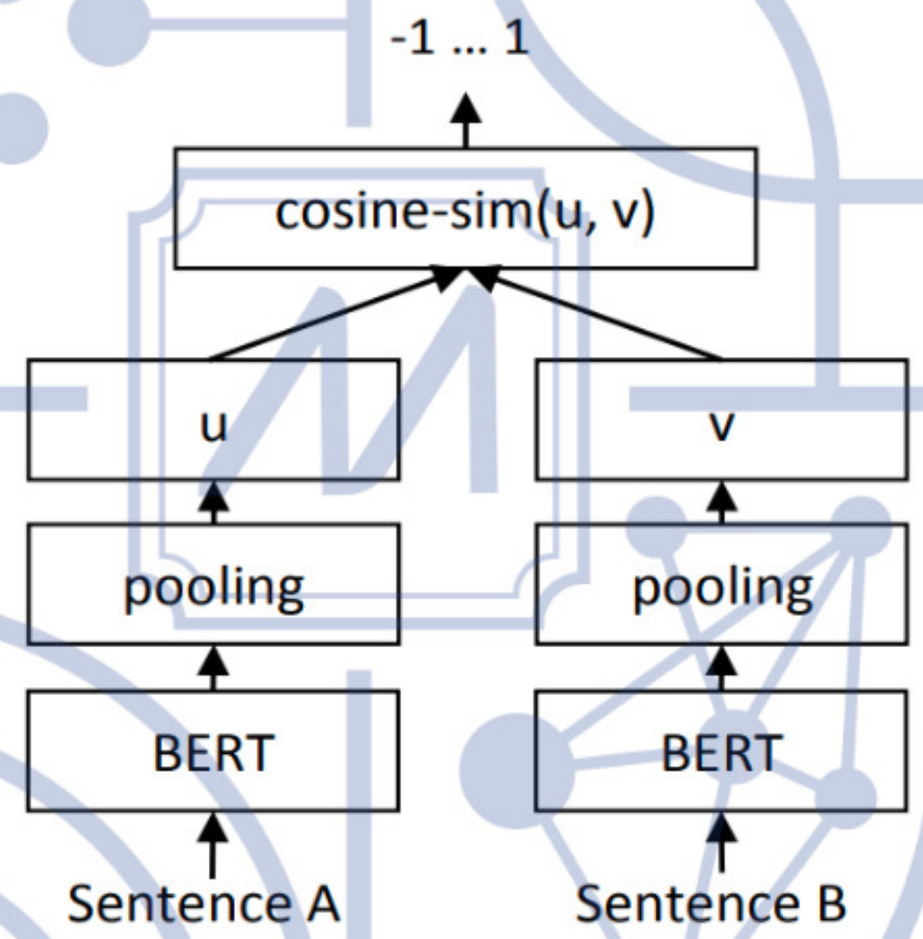
3. $u \cdot v$= dot product (perkalian titik) antara $u$dan v

4. $\| u \|, \| v \|$= panjang (norma) dari vektor $u$dan v

di mana $u = SBERT(S_1)$dan $v = SBERT(S_2)$. Nilai *similarity* berada pada rentang -1 hingga 1, di mana nilai yang lebih tinggi menunjukkan tingkat kemiripan semantik yang lebih besar [23].

2.6.3 Arsitektur *Siamese network* pada S*BERT*

Arsitektur S*BERT* menggunakan struktur *siamese network*, yaitu dua jaringan *BERT* identik yang berbagi parameter. Setiap kalimat diproses secara terpisah melalui *encoder BERT* untuk menghasilkan representasi token-level, dan diringkas menjadi representasi kalimat melalui proses *pooling* [41], [42], [44].



Gambar 2. 1 Arsitektur siamese network pada sbert [45].

Pooling yang umum digunakan meliputi:

1. Mean Pooling yang bekerja dengan cara menghitung rata-rata aritmatika dari seluruh vektor *embedding* token yang valid di dalam sebuah kalimat. Secara matematis, setiap dimensi pada vektor hasil dihitung dengan menjumlahkan nilai dimensi tersebut dari seluruh token, kemudian dibagi dengan jumlah total token. Pendekatan ini dinilai efektif karena mampu menangkap keseluruhan makna semantik dari kalimat dengan mempertimbangkan kontribusi setiap kata, serta lebih tahan terhadap derau (*noise*) pada token tertentu [41], [44].

2. Max Pooling yang bekerja dengan cara mengambil nilai maksimum dari setiap dimensi vektor di sepanjang seluruh token pada kalimat [41]. Untuk setiap dimensi, sistem akan memilih nilai terbesar yang dihasilkan oleh token mana pun dalam urutan tersebut. Tujuan dari pendekatan ini adalah untuk menangkap fitur atau sinyal paling menonjol (*salient*

features) yang muncul dalam kalimat [44]. Meskipun mampu menangkap kata-kata kunci yang kuat, *max pooling* berpotensi kehilangan informasi struktural dan konteks keseluruhan kalimat karena hanya berfokus pada nilai puncak tertentu di setiap dimensi.

3. CLS Pooling yang menggunakan representasi dari token khusus [CLS] (Classification) yang berada pada posisi pertama dari setiap kalimat [38]. Pada arsitektur BERT standar, token [CLS] dilatih secara khusus selama proses *pre-training* untuk merangkum informasi global dari seluruh kalimat untuk keperluan tugas klasifikasi [38], [41]. Dalam metode *CLS Pooling*, vektor *embedding* dari token [CLS] pada lapisan terakhir diambil secara langsung sebagai representasi tunggal untuk keseluruhan kalimat. Pendekatan ini seringkali kurang optimal untuk tugas pengukuran kemiripan semantik (*semantic similarity*) dibandingkan dengan *mean pooling* [41].

Dari proses tersebut diperoleh *embedding* kalimat berdimensi tetap yang siap digunakan untuk perhitungan kemiripan [41].

Pada tahap pelatihan, S*BERT* menggunakan dataset *Natural Language Inference (NLI)* dan *Semantic Textual Similarity (STS)* untuk mempelajari hubungan semantik antar kalimat. Fungsi objektif yang digunakan dapat berupa classification loss (pada NLI) atau regression loss (pada STS), sehingga model mampu membedakan pasangan kalimat yang serupa maupun berbeda secara makna [41].

2.6.4 Alur Kerja S*BERT* dalam Pemrosesan Teks

Proses kerja *SBERT* dalam sistem berbasis kemiripan teks dapat dijelaskan sebagai berikut [38]:

1. *SBERT* menggunakan tokenizer *BERT* berbasis *WordPiece*, yang memecah kalimat menjadi *subword* token [37]. Setiap kalimat dibungkus dengan token khusus. Token ID kemudian dipetakan ke indeks dalam kosakata *BERT* (vocab size = 30.522 untuk *BERT*-base). Panjang maksimum sekuens adalah 512 token.
2. Setiap token direpresentasikan sebagai penjumlahan tiga jenis *embedding*:

$$\mathbf{E}_{input} = \mathbf{E}_{token} + \mathbf{E}_{position} + \mathbf{E}_{segment} \quad \text{.....(4)}$$

Keterangan:

$\mathbf{E}_{token}$ = representasi makna leksikal token

$\mathbf{E}_{position}$ = posisi token

$\mathbf{E}_{segment}$ = pembeda kalimat A atau B

3. *Embedding* diproses melalui L lapisan transformer. Setiap lapisan menerapkan *Multi-Head Self-attention* dan *Feed-Forward Network*:

Self-attention per head:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \quad \text{.....(5)}$$

Keterangan:

1. $Q = XW_Q, K = XW_K, V = XW_V$(matriks proyeksi yang dipelajari)
2. d_k= dimensi setiap head

Multi-Head Attention:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

Keterangan:

1. Q= *Query* (representasi yang mencari informasi)
2. K= *Key* (representasi acuan pencocokan)
3. V= *Value* (informasi yang akan diambil)
4. $\text{head}_1, \ldots, \text{head}_h$= hasil attention dari masing-masing head
5. h= jumlah head pada *Multi-Head attention*
6. W^O= matriks bobot *(weight)* untuk transformasi *output*

Output**:** Matriks kontekstual $\mathbf{H} \in \mathbb{R}^{n \times d}$,

Keterangan:

1. H= hasil akhir attention
2. n= jumlah token
3. d= dimensi *embedding*
4. $\mathbb{R}^{n \times d}$= matriks berukuran $n \times d$(n baris, d kolom)

4. Karena output *BERT* menghasilkan *embedding* per-token, diperlukan operasi *pooling* untuk menghasilkan vector kalimat berdimensi tetap. S*BERT* mendukung tiga strategi:

1. *Mean Pooling*

$$u = \frac{1}{n}\sum_{i=1}^{n} H_i \quad \text{.....(6)}$$

Keterangan:

1. u= vektor hasil (representasi kalimat)
2. n= jumlah token
3. H_i= *embedding* token ke-i

2. *Max Pooling*

$$u_j = \max_i(H_{ij}) \quad (7)$$

Keterangan:

1. u_j= nilai pada dimensi ke-j dari vektor hasil
2. H_{ij}= nilai *embedding* token ke-i pada dimensi ke-j

3. *CLS Pooling*

$$u = H_{[CLS]} \quad (8)$$

Keterangan:

1. u= vektor hasil (representasi kalimat)
2. $H_{[CLS]}$= *embedding* token [CLS]

Penelitian Reimers & Gurevych (2019) menunjukkan bahwa mean *pooling* secara konsisten menghasilkan kinerja terbaik pada *benchmark Semantic Textual Similarity*.

$$u = \text{MeanPool}(H_1, H_2, \dots, H_n) \in \mathbb{R}^{768} \quad (9)$$

Keterangan:

1. u= vektor hasil (representasi kalimat)
2. $H_1, H_2, \dots, H_n$= *embedding* dari setiap token
3. n= jumlah token dalam kalimat
4. $\mathbb{R}^{768}$= ruang vektor berdimensi 768

5. Hasil *pooling* menghasilkan satu vektor per kalimat dengan dimensi tetap $d = 768$. Vektor ini merupakan representasi semantik padat dari seluruh kalimat:

$$\text{sentence\_embedding}(s) = \text{SBERT}(s) \in \mathbb{R}^{768} \quad (10)$$

Keterangan:

1. s= kalimat (input teks)
2. sentence_embedding(s)= representasi vektor dari kalimat s
3. SBERT(s)= proses en*coding* kalimat menggunakan model *Sentence-BERT*
4. $\mathbb{R}^{768}$= ruang vektor berdimensi 768

Embedding ini dapat dihitung sekali dan disimpan, menjadikan S*BERT* sangat efisien untuk pencarian skala besar.

6. Kemiripan antara dua kalimat $s_1$dan $s_2$dihitung menggunakan *cosine similarity* antara *embedding* mereka:

$$\cos_{\text{sim}}(u, v) = \frac{u \cdot v}{\|u\| \, \|v\|} \quad \text{.....(11)}$$

Keterangan:

1. u, v= dua vektor
2. $u \cdot v$= *dot product* (perkalian titik) antara $u$dan v
3. $\| u \|, \| v \|$= panjang (norma) dari vektor $u$dan v

Pendekatan ini memungkinkan sistem melakukan pencarian atau penilaian teks secara cepat karena *embedding* dapat dihitung satu kali dan digunakan berulang.

2.6.5 Keunggulan dan Aplikasi S*BERT*

S*BERT* memiliki keunggulan utama dalam efisiensi dan kualitas representasi semantik dibandingkan *BERT* standar. Dengan pendekatan *embedding* independen, kompleksitas perbandingan teks dapat dikurangi secara signifikan dari orde kuadrat menjadi linear dan sangat cocok untuk aplikasi skala besar [41]. S*BERT* juga mampu menghasilkan *embedding* yang lebih stabil dan representatif untuk tugas *semantic similarity*, dan banyak digunakan dalam berbagai aplikasi *NLP* modern, antara lain:

1. pencarian informasi berbasis semantik
2. klasterisasi dokumen
3. deteksi parafrase
4. sistem rekomendasi teks
5. *Automatic Essay Scoring*

Kemampuan tersebut menjadikan S*BERT* sebagai salah satu metode yang paling efektif dalam sistem yang memerlukan pemahaman makna teks secara mendalam dan efisien, khususnya pada sistem evaluasi jawaban esai berbasis kesamaan semantic [42].

2.7 Cosine Similarity

Cosine similarity merupakan metode pengukuran kemiripan yang banyak digunakan dalam *Natural Language Processing*, khususnya dalam sistem *Automatic Essay Scoring*

berbasis *embedding* seperti S*BERT*. Metode ini digunakan untuk mengukur kedekatan makna semantik antara dua teks dengan membandingkan representasi vektornya dalam ruang berdimensi tinggi. *Cosine similarity* bekerja dengan menghitung nilai *cosinus* dari sudut yang terbentuk antara dua vektor, sehingga fokus pada arah vektor. Nilai *cosine similarity* berada pada rentang -1 hingga 1, di mana nilai 1 menunjukkan kedua vektor identik, nilai 0 menunjukkan tidak adanya hubungan (ortogonal), dan nilai -1 menunjukkan arah yang berlawanan [46], [47]. Secara matematis, *cosine similarity* antara dua vektor Adan Bdirumuskan sebagai:

$$CosineSimilarity = 1 - \frac{A.B}{||A||.||B||} \quad \text{.....(12)}$$

Keterangan:

1. A, B= dua vektor
2. $A \cdot B$= perkalian titik antara vektor Adan B
3. $\| A \|, \| B \|$= panjang dari vektor Adan B

Pendekatan ini memiliki keunggulan karena tidak dipengaruhi oleh panjang teks dan sangat efektif digunakan untuk membandingkan dokumen atau kalimat dengan ukuran yang berbeda [44], [46]. Dalam sistem penilaian esai otomatis, *cosine similarity* digunakan untuk menghitung tingkat kemiripan antara *embedding* jawaban mahasiswa dan kunci jawaban, dan dikonversi menjadi skor numerik sebagai hasil evaluasi. Dengan kemampuannya dalam menangkap kesamaan semantik secara kuantitatif dan efisien, *cosine similarity* menjadi komponen penting dalam sistem evaluasi berbasis teks modern [46], [47], [48].

2.8 Root Mean Square Error (RMSE)

Evaluasi Root Mean Square Error (RMSE) merupakan metrik evaluasi yang digunakan untuk mengukur sejauh mana skor yang diprediksi oleh sistem penilaian esai otomatis mendekati skor yang diberikan oleh penilai manusia. *RMSE* dihitung dari akar kuadrat rata-rata selisih kuadrat antara nilai aktual dan nilai prediksi, sehingga memberikan penalti lebih besar terhadap kesalahan prediksi yang jauh dari nilai sebenarnya dan mendorong model untuk menghasilkan prediksi yang lebih akurat dan konsisten di seluruh rentang nilai [46], [47]. Secara matematis, *RMSE* dinyatakan sebagai:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (yi - \hat{y}i)^2} \quad \text{.....(13)}$$

Keterangan:

1. n= jumlah data/sampel

2. Y_i= nilai aktual pada data ke-i
3. $\hat{Y}_i$= nilai prediksi pada data ke-i

Nilai *RMSE* yang lebih kecil menunjukkan tingkat kesalahan yang rendah dan menandakan bahwa sistem mampu memprediksi skor dengan lebih akurat dibandingkan nilai referensi [47]. *RMSE* menjadi metrik yang tepat untuk menilai performa sistem *Automatic Essay Scoring*, karena mengkuantifikasi perbedaan antara prediksi model dan penilaian manusia secara objektif dan konsisten [21], [47].

2.9 Penelitian Terdahulu

Penelitian mengenai *Automatic Essay Scoring* terus berkembang seiring meningkatnya kebutuhan evaluasi pembelajaran berbasis teknologi, khususnya dalam era pendidikan digital yang menuntut kecepatan, konsistensi, dan skalabilitas dalam proses penilaian. Tinjauan terhadap berbagai penelitian terdahulu berikut ini menggambarkan perkembangan pendekatan dan metode yang digunakan, sekaligus menjadi landasan bagi penelitian yang sedang dikembangkan.

Penelitian Wang et al. (2022) berjudul *"On the Use of BERT for Automated Essay Scoring: Joint Learning of Multi-Scale Essay Representation"* merupakan salah satu kontribusi penting dalam penggunaan *BERT* untuk *AES* yang dipresentasikan pada *konferensi NAACL* 2022 [26]. Penelitian ini mengusulkan representasi esai *multi*-skala yang secara bersama-sama dipelajari oleh model *BERT*, mencakup representasi di tingkat token, segmen, dan dokumen secara sekaligus. Kontribusi utama penelitian ini adalah menunjukkan bahwa *BERT* yang *dikonfigurasi* dengan benar menggunakan representasi multi-skala dan fungsi loss ganda dapat melampaui performa model berbasis *LSTM* yang sebelumnya mendominasi tugas *AES*. Model yang diusulkan berhasil mencapai hasil yang hampir setara state-of-the-art pada dataset ASAP dengan rata-rata *Quadratic Weighted Kappa (QWK)* sekitar 0,791, sekaligus menunjukkan generalisasi yang baik pada dataset *CommonLit Readability Prize* [26].

Penelitian Wu et al. (2023) berjudul "*Beyond Benchmarks: Spotting Key Topical Sentences While Improving Automated Essay Scoring Performance with Topic-Aware BERT"* yang dipublikasikan dalam jurnal *Electronics* memberikan kontribusi ganda, yaitu meningkatkan akurasi penilaian sekaligus menambahkan kemampuan interpretabilitas [49]. Penelitian ini mengusulkan metode *Topic-Aware BERT* yang menyertakan informasi topik dari instruksi penulisan esai sebagai masukan tambahan, bukan hanya teks esai itu sendiri. Dengan menambahkan informasi topik, model dapat lebih memahami relevansi jawaban

terhadap pertanyaan yang diberikan. Hasil eksperimen pada dataset *ASAP* menunjukkan bahwa *Topic-Aware BERT* dengan ekstraksi kalimat topik otomatis menggunakan metode *Xsum* mencapai QWK rata-rata 0,789, meningkat dibandingkan *Vanilla BERT* yang hanya mencapai 0,774 [49] . model ini juga mampu mengidentifikasi kalimat-kalimat kunci dalam esai yang paling relevan dengan topik, memberikan informasi yang berguna untuk umpan balik kepada peserta didik.

Penelitian Chamidah et al. (2022) berjudul "Penilaian Esai Pendek Otomatis Berdasarkan *Similaritas Semantik* dengan *SBERT*" yang dipublikasikan dalam jurnal Techno.COM merupakan salah satu penelitian penting dalam pengembangan *AES* berbahasa Indonesia menggunakan S*BERT*. Penelitian ini mengembangkan sistem penilaian esai pendek otomatis di mana teks jawaban dan kunci jawaban masing-masing direpresentasikan sebagai vektor 512 dimensi menggunakan model S*BERT*, kemudian dibandingkan menggunakan *cosine similarity* untuk menghasilkan skor esai. Evaluasi terhadap model dilakukan menggunakan Mean Absolute Error (MAE) dan Pearson Correlation dengan nilai referensi dari penilai manusia. Hasil penelitian menunjukkan rata-rata MAE sebesar 0,26 dan rata-rata korelasi Pearson sebesar 0,78, yang menandakan tingkat kesesuaian yang cukup tinggi antara skor sistem dan skor penilai manusia. Penelitian ini membuktikan bahwa representasi semantik berbasis S*BERT* lebih unggul dibandingkan metode berbasis pencocokan kata kunci tradisional untuk penilaian esai pendek dalam Bahasa Indonesia.

Penelitian Susanto et al. (2023) berjudul *"Performance of Text Similarity Algorithms for Essay Answer Scoring in Online Examinations"* yang dipublikasikan dalam Jurnal Teknik Informatika *(JUTIF)* melakukan analisis komparatif terhadap berbagai algoritma text *similarity* dalam konteks penilaian jawaban esai pada sistem ujian daring. Penelitian ini menguji lima algoritma similarity yang berbeda termasuk *TF-IDF*, Jaccard, dan *LSA (Latent Semantic Analysis)* menggunakan 371 teks jawaban dari sistem ujian *daring*. *Evaluasi* dilakukan menggunakan *RMSE* dengan nilai referensi penilaian manusia. Hasil penelitian menunjukkan bahwa algoritma *TF-IDF* berbasis Jaccard menghasilkan *RMSE* terendah dibandingkan penilaian manusia, namun algoritma *LSA* menunjukkan kemampuan lebih baik dalam mengikuti pola penilaian manusia untuk soal esai deskriptif. Temuan ini menegaskan bahwa tidak ada satu algoritma yang optimal untuk semua kasus, dan pemilihan metode harus disesuaikan dengan karakteristik dataset dan jenis pertanyaan.

Penelitian Pradani dan Suadaa (2023) berjudul *"Automated Essay Scoring Menggunakan Semantic Textual Similarity Berbasis Transformer untuk Penilaian Ujian Esai"* yang dipublikasikan dalam Jurnal Teknologi Informasi dan Ilmu Komputer *(JTIIK)*

menerapkan model berbasis Transformer untuk sistem *AES* berbahasa Indonesia di Politeknik Statistika *STIS*. Penelitian ini menggunakan model *fine-tuned* Indo*BERT* sebagai model utama dengan *TF-IDF cosine similarity* sebagai baseline, kemudian mengkonversi skor kemiripan menjadi skor nilai dalam rentang yang ditentukan. Hasil evaluasi menunjukkan bahwa model fine-tuned Indo*BERT* menghasilkan performa terbaik dengan nilai *MAE* sebesar 0,1285 dan RMSE sebesar 0,2001, yang jauh lebih baik dibandingkan baseline *TF-IDF cosine similarity*. Penelitian ini menjadi salah satu referensi penting yang menunjukkan bahwa pendekatan berbasis Transformer dapat diadaptasi dengan baik untuk penilaian esai berbahasa Indonesia.

Penelitian Permata et al. (2025) berjudul *"Towards an Automated Essay Evaluation System NLP Based Text Embeddings and Similarity Metrics"* yang dipublikasikan dalam Jurnal Digital *Zone* mengembangkan sistem evaluasi jawaban esai berbasis *NLP* dengan memanfaatkan berbagai teknik text *embedding* dan metode cosine similarity. Sistem yang dikembangkan memungkinkan evaluasi jawaban esai secara otomatis yang dapat terintegrasi dengan platform pembelajaran daring. Hasil penelitian menunjukkan bahwa pendekatan berbasis *embedding* teks dan *cosine similarity* mampu menghasilkan skor penilaian yang stabil dan konsisten, dengan tingkat kesalahan yang dapat diterima untuk keperluan evaluasi pendidikan.

Penelitian Yuliyanti dan Ula (2021) berjudul *"Essay Answer Detection System Uses Cosine similarity and Similarity Scoring in Sentences"* yang dipublikasikan dalam International Journal of Applied Science and Smart Technology mengembangkan sistem deteksi kemiripan jawaban esai berbasis *cosine similarity* dengan representasi vektor multidimensi [47]. Penelitian ini menerapkan *cosine similarity* langsung pada representasi TF-IDF dari jawaban esai dan kunci jawaban untuk menghasilkan skor kemiripan, yang kemudian dikonversi menjadi skor penilaian. Sistem yang dikembangkan diuji pada dataset jawaban esai pendek dalam Bahasa Indonesia dan menunjukkan kemampuan yang cukup baik dalam mengukur tingkat kesamaan jawaban [47].

Penelitian Nie (2025) berjudul "Automated Essay Scoring with S*BERT Embeddings* and LSTM-Attention Networks" yang dipublikasikan dalam jurnal *PeerJ Computer Science* merupakan salah satu penelitian terbaru yang menggabungkan kekuatan S*BERT* dengan kapabilitas pemodelan sekuensial *LSTM* untuk meningkatkan akurasi *AES* [42]. Penelitian ini mengusulkan arsitektur hibrid di mana S*BERT* digunakan untuk menghasilkan *embedding* yang kaya makna semantik dari setiap esai, kemudian diproses oleh lapisan BiLSTM (*Bidirectional* LSTM) dengan mekanisme attention untuk menangkap pola

sekuensial dan fitur penting dalam *embedding* tersebut. Evaluasi komprehensif menggunakan berbagai metrik menunjukkan bahwa model ini mencapai *QWK* sebesar 0,7876 pada dataset benchmark *ASAP*, melampaui performa model SkipFlow *(QWK 0,6820)* dan *XLNet (QWK 0,7042)* [42]. Penelitian ini membuktikan bahwa kombinasi *embedding* semantik S*BERT* dengan pemodelan sekuensial *LSTM* dapat menghasilkan performa *AES* yang lebih baik dibandingkan penggunaan salah satu model secara sendiri.

Penelitian [43] berjudul "Penerapan Sentence-*BERT* dan *Cosine similarity* untuk Pencarian Semantik Dokumen Skripsi dalam Format PDF" yang dipublikasikan dalam jurnal Ranah Research mengembangkan sistem pencarian semantik untuk dokumen skripsi berbahasa Indonesia menggunakan S*BERT* dan *cosine similarity* [43]. Meskipun fokus penelitian ini adalah pencarian semantik dan bukan penilaian esai, penelitian ini memberikan kontribusi penting dalam memvalidasi efektivitas S*BERT* dan *cosine similarity* untuk pemrosesan teks akademik berbahasa Indonesia. Sistem yang dikembangkan menunjukkan kemampuan yang signifikan dalam memahami hubungan makna antar kalimat dan dokumen berbahasa Indonesia, bahkan ketika menggunakan kata-kata yang berbeda namun bersinonim [43].

Penelitian Nasirin dan Indahsari (2025) berjudul "Model Penilaian Jawaban Esai Berbasis *Semantic Understanding* Menggunakan *LLaMa* dan *Text Similarity*" yang dipublikasikan dalam Jurnal Edukasi dan Penelitian Informatika mengembangkan sistem penilaian esai yang menggabungkan tiga komponen utama, yaitu *SentenceTransformer* untuk representasi semantik, *TF-IDF* untuk representasi berbasis frekuensi kata, dan model *LLaMa (Large Language Model)* sebagai komponen pemahaman semantik tingkat tinggi. Penelitian ini mengusulkan pendekatan *hybrid* yang memanfaatkan kelebihan masing-masing metode, di mana *TF-IDF* menangkap kemiripan berbasis kata kunci, *SentenceTransformer* menangkap kemiripan semantik, dan *LLaMa* memberikan pemahaman kontekstual yang lebih mendalam. Kombinasi ketiga metode

Tabel 2. 1 Ringkasan Penelitian Terdahulu

No	Peneliti	Tahun	Metode	Dataset	Hasil	Kelebihan	Kekurangan
1	Wang et al.	2022	*BERT* Multi-Scale Essay	ASAP Dataset	Rata-rata QWK ≈ 0,791 pada dataset ASAP;	Pendekatan multi-skala komprehensif; melampaui	Hanya diuji pada dataset Bahasa Inggris; kompleksitas

			Representation		melampaui performa LSTM	performa LSTM; generalisasi baik pada dua dataset berbeda	arsitektur tinggi; belum ada antarmuka pengguna
2	Wu et al.	2023	Topic-Aware *BERT*	ASAP Dataset	QWK rata-rata 0,789 (Xsum-T *BERT*), meningkat dari Vanilla *BERT* 0,774	Meningkatkan interpretabilitas dengan identifikasi kalimat kunci; menambahkan konteks topik	Bergantung pada kualitas ekstraksi kalimat topik; performa bervariasi tergantung metode ekstraksi
3	Chamidah et al.	2022	S*BERT* + Cosine Similarity	Short Answer Dataset (Bahasa Indonesia)	Rata-rata MAE = 0,26; Pearson Correlation = 0,78	Menggunakan S*BERT* untuk Bahasa Indonesia; hasil korelasi tinggi (0.78); relatif sederhana dan efisien	Tidak berbasis website; hanya mengevaluasi esai pendek; tidak ada antarmuka pengguna
4	Susanto et al.	2023	Komparasi TF-IDF, Jaccard, LSA, dll.	371 teks ujian daring	TF-IDF Jaccard: RMSE terendah; LSA: paling mengikuti pola penilaian manusia	Analisis komparatif komprehensif 5 algoritma; menggunakan dataset ujian daring nyata	Tidak menggunakan pendekatan *deep learning*; tidak ada antarmuka pengguna terintegrasi

5	Pradani & Suadaa	2023	Indo*BERT* fine-tuned + Cosine Similarity	Essay Dataset Bahasa Indonesia (STIS)	Indo*BERT*: MAE = 0,1285; RMSE = 0,2001 (terbaik vs TF-IDF baseline)	Menggunakan Indo*BERT* fine-tuned untuk Bahasa Indonesia; MAE dan RMSE sangat rendah	Memerlukan data fine-tuning berlabel; tidak berbasis website; hanya diuji pada satu institusi
6	Permata et al.	2025	*NLP* Text *Embedding* + Cosine Similarity	Essay Answer Dataset	Sistem menghasilkan skor penilaian otomatis yang stabil dan konsisten	Mendukung integrasi platform e-learning; pendekatan *embedding* fleksibel	Detail metrik evaluasi tidak disebutkan secara spesifik; tidak ada perbandingan dengan model lain
7	Yuliyanti & Ula	2021	*Cosine similarity* + Similarity Scoring	Short Essay Dataset Bahasa Indonesia	Sistem mampu mengukur kemiripan jawaban esai pendek dengan baik	Implementasi *cosine similarity* langsung; mudah dipahami dan diimplementasikan	Hanya menggunakan TF-IDF; tidak menangkap makna semantik; akurasi terbatas
8	Nie	2025	S*BERT* + BiLSTM + Attention	ASAP Benchmark Dataset	QWK = 0,7876; melampaui SkipFlow (0,6820) dan XLNet (0,7042)	Kombinasi S*BERT* + BiLSTM + Attention menghasilkan QWK tertinggi; evaluasi komprehensif	Arsitektur lebih kompleks; tidak diuji pada Bahasa Indonesia; tidak ada antarmuka pengguna

9	Fathuddin et al.	2024	S*BERT* + *Cosine similarity* (semantic search)	Dokumen Skripsi PDF Bahasa Indonesia	Sistem mampu memahami hubungan makna antar dokumen berbahasa Indonesia secara efektif	Memvalidasi S*BERT* untuk teks akademik Bahasa Indonesia; mendukung pemrosesan PDF	Fokus pencarian semantik bukan penilaian esai; tidak menghasilkan skor numerik
10	Nasirin & Indahsari	2025	SentenceTransformer + TF-IDF + LLaMa	Essay Dataset Bahasa Indonesia	Kombinasi hybrid menghasilkan penilaian lebih robust terhadap variasi gaya penulisan	Pendekatan hybrid tiga metode lebih robust; memanfaatkan *LLM* untuk pemahaman kontekstual	Kompleksitas sistem tinggi; ketergantungan pada model LLaMa yang berat secara komputasi

ini menghasilkan sistem penilaian yang lebih *robust* terhadap variasi gaya penulisan mahasiswa dan mampu memberikan penilaian yang lebih *komprehensif* dibandingkan pendekatan *single-method.*

Berdasarkan Tabel 2.2, dapat disimpulkan bahwa penelitian-penelitian terdahulu telah menunjukkan perkembangan signifikan dalam pengembangan sistem *AES* menggunakan berbagai pendekatan *NLP* dan *deep learning*. Penelitian berbasis *Transformer* seperti *BERT* dan variannya secara konsisten menghasilkan performa yang lebih baik dibandingkan metode tradisional berbasis TF-IDF, dengan nilai QWK yang mencapai 0,79 ke atas pada dataset benchmark internasional. Untuk konteks Bahasa Indonesia, penggunaan model pra-latih seperti Indo*BERT* dan S*BERT* menunjukkan hasil yang menjanjikan dengan nilai *MAE* dan *RMSE* yang lebih rendah dibandingkan metode *konvensional*, memvalidasi relevansi pendekatan berbasis Transformer untuk penilaian esai dalam Bahasa Indonesia.

Penelitian ini memposisikan diri sebagai pengembangan lanjutan dari penelitian Chamidah et al. (2022) dan Pradani & Suadaa (2023) dengan memanfaatkan Sentence-*BERT* dan *cosine similarity* untuk penilaian esai berbahasa Indonesia, namun menambahkan antarmuka berbasis website yang memudahkan penggunaan sistem dalam konteks

pendidikan daring. Berbeda dengan penelitian Nie (2025) yang menambahkan komponen *LSTM* setelah S*BERT* untuk meningkatkan akurasi, penelitian ini berfokus pada kesederhanaan dan efisiensi implementasi tanpa mengorbankan akurasi penilaian semantik, dengan evaluasi menggunakan *RMSE* terhadap skor referensi penilai manusia.