

**PENGEMBANGAN DATASET BERBAHASA INDONESIA UNTUK
AUTOMATED ESSAY SCORING (AES) DENGAN FITUR
FEEDBACK MELALUI IMPLEMENTASI TEKNIK
FINE-TUNING BERT PADA
PLATFORM *WEBSITE***

SKRIPSI

STEPHEN J. RUSLI

211401059



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2025

UNIVERSITAS SUMATERA UTARA

**PENGEMBANGAN DATASET BERBAHASA INDONESIA UNTUK
AUTOMATED ESSAY SCORING (AES) DENGAN FITUR
FEEDBACK MELALUI IMPLEMENTASI TEKNIK
FINE-TUNING BERT PADA
PLATFORM *WEBSITE***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer

STEPHEN J. RUSLI

211401059



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2025

PERSETUJUAN

Judul : PENGEMBANGAN DATASET BERBAHASA
INDONESIA UNTUK *AUTOMATED ESSAY
SCORING* (AES) DENGAN FITUR *FEEDBACK*
MELALUI IMPLEMENTASI TEKNIK *FINE-
TUNING* BERT PADA PLATFORM *WEBSITE*

Kategori : SKRIPSI

Nama : STEPHEN J. RUSLI

Nomor Induk Mahasiswa : 211401059

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Tanggal Sidang : 09 JANUARI 2025

Komisi Pembimbing :
Pembimbing 2



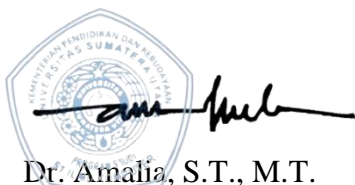
Sri Melvani Hardi, S.Kom., M.Kom.
NIP. 198805012015042006

Pembimbing 1



Dr. Amalia, S.T., M.T.
NIP. 197812212014042001

Diketahui/Disetujui Oleh
Program Studi S-1 Ilmu Komputer
Ketua,



Dr. Amalia, S.T., M.T.
NIP. 197812212014042001

PERNYATAAN

**PENGEMBANGAN DATASET BERBAHASA INDONESIA UNTUK
AUTOMATED ESSAY SCORING (AES) DENGAN FITUR
FEEDBACK MELALUI IMPLEMENTASI TEKNIK
FINE-TUNING BERT PADA
PLATFORM *WEBSITE***

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 19 November 2024



Stephen J. Rusli

211401059

PENGHARGAAN

Puji syukur penulis panjatkan ke Tuhan yang Maha Esa, atas karunia-Nya, penulis dapat menuntaskan penyusunan skripsi ini sebagai syarat untuk mendapatkan gelar Sarjana Komputer pada Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Selama penulis menyusun skripsi ini, tentunya tidak lepas dari bimbingan dan dukungan berbagai pihak. Oleh sebab itu, penulis ingin menyampaikan tanda terima kasih kepada:

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan sebagai Dosen Pembimbing I yang telah memberikan banyak masukan, motivasi, serta dukungan kepada penulis selama ini.
5. Ibu Sri Melvani Hardi, S.Kom., M.Kom. selaku Sekretaris Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan sebagai Dosen Pembimbing II yang telah memberikan saran kepada penulis selama ini.
6. Ibu Hayatunnufus S.Kom, M.Cs selaku Dosen Pembimbing Akademik yang telah memberikan banyak arahan kepada penulis selama perkuliahan.
7. Seluruh Bapak dan Ibu Dosen Program Studi S-1 Ilmu Komputer, yang telah membimbing penulis selama masa perkuliahan.
8. Teristimewa kepada Orang Tua terkasih, tersayang penulis Drs. Jimmy Rusli, S.E. dan Peirani Salim yang telah memberikan penulis kasih sayang yang tiada henti, ilmu yang bermanfaat, dan dukungan yang luar biasa hingga penulis dapat menjalani perkuliahan dengan baik sampai dengan penyusunan skripsi.
9. Kakak kandung tercinta Stepheny J. Rusli yang selalu mendukung dan mendoakan penulis dalam menjalani kehidupan masa kuliah hingga sampai menyelesaikan

tugas akhir.

10. Asisten Laboratorium IKLC USU stambuk 2020 sampai 2023 yang membimbing dan menemani penulis selama ini.
11. Pengurus IMILKOM Tahun 2023/2024, terkhusus BPH dan seluruh pengurus Departemen Wawasan Kontemporer yang telah banyak membantu penulis dalam setiap kegiatan di lingkungan organisasi IMILKOM.
12. Sahabat penulis yaitu Akhdan, NY, Alex, Harry, Husein, Rizky, Lorenzo, Hansen, Zaki, Faiz, Iqbal, Riyan, Sutri, Sea, Belvin, Caca yang telah menemani dan mendukung penulis selama perkuliahan.
13. Stambuk 2021 terkhusus kom A yang telah menemani penulis selama masa studi dan memberikan banyak kenangan tak terlupakan.
14. Serta seluruh pihak yang turut berkontribusi, yang tidak dapat penulis sebutkan satu per-satu.

Medan, 19 November 2024

Penulis,



Stephen J. Rusli

ABSTRAK

Natural Language Processing (NLP) merupakan bidang ilmu *artificial intelligence* (AI) yang memfasilitasi bagaimana komputer dapat berinteraksi dengan manusia menggunakan bahasa yang kita pahami, baik dalam teks atau suara. Salah satu pengaplikasian dari NLP adalah *Automated Essay Scoring* (AES), yang bertujuan untuk memberikan penilaian otomatis terhadap esai dengan akurasi yang tepat. Penelitian ini bertujuan untuk membangun model AES berbasis *Transformer*, spesifiknya menggunakan model IndoBERT yang dirancang khusus untuk Bahasa Indonesia. Dataset pelatihan dibuat secara manual untuk memastikan data yang akurat dan relevan, sekaligus menghindari potensi kesalahan makna yang sering terjadi akibat proses terjemahan dari bahasa asing. Model dirancang untuk dapat menghasilkan penilaian berdasarkan beberapa rubrik penilaian, yakni relevansi jawaban dengan *prompt*, korelasi antar kalimat, panjang esai, dan kekayaan kosakata. Tahap terakhir yaitu evaluasi, memanfaatkan metrik *Mean Squared Error* (MSE) mendapatkan nilai sebesar 0.003 serta menggunakan *Quadratic Weighted Kappa* (QWK) mendapatkan nilai sebesar 0.92. Hasil evaluasi ini memberi kesimpulan bahwa penilaian yang dihasilkan model cukup akurat dan selaras dengan standar penilaian manual. Dengan adanya penelitian ini, diharapkan dapat menjadi awalan yang bagus dalam pengembangan sistem penilaian esai otomatis berbasis Bahasa Indonesia dan dapat diimplementasikan untuk berbagai keperluan di bidang pendidikan.

Kata Kunci: *Dataset, Automatic Essay Scoring, Natural Language Processing, BERT, Fine-Tuning, Feedback*

ABSTRACT

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that facilitates how computers can interact with humans using language that we understand, either in text or voice. One of the applications of NLP is Automated Essay Scoring (AES), which aims to provide automatic scoring of essays with the right accuracy. This research aims to build a Transformer-based AES model, specifically using the IndoBERT model specifically designed for Indonesian. The training dataset is manually created to ensure accurate and relevant data, while avoiding potential meaning errors that often occur due to the translation process from foreign languages. The model is designed to be able to generate scoring based on several scoring rubrics, namely the relevance of the answer to the prompt, correlation between sentences, essay length, and vocabulary richness. The last stage is evaluation, utilizing the Mean Squared Error (MSE) metric to get a value of 0.003 and using Quadratic Weighted Kappa (QWK) to get a value of 0.92. The results of this evaluation conclude that the assessment produced by the model is quite accurate and in line with the standard manual assessment. With this research, it is hoped that it can be a good start in developing an automatic essay grading system based on Indonesian and can be implemented for various purposes in the field of education.

Keywords: *Dataset, Automatic Essay Scoring, Natural Language Processing, BERT, Fine-Tuning, Feedback*

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN.....	iv
PENGHARGAAN.....	v
ABSTRAK	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
1.6 Metodologi Penelitian	5
1.7 Penelitian Relevan	6
1.8 Sistematika Penulisan.....	7
BAB II LANDASAN TEORI	8
2.1 Dataset.....	8
2.2 <i>Automated Essay Scoring</i>	8
2.3 <i>Deep Learning</i>	9
2.3.1 <i>Transformers</i>	9
2.3.2 <i>Hyperparameter</i>	10
2.3.3 <i>Learning Rate</i>	10
2.3.4 <i>Batch Size</i>	11
2.3.5 <i>Activation Function</i>	11
2.4 <i>Natural Language Processing</i>	12
2.4.1 Tokenisasi	13
2.4.2 Penghapusan Karakter Khusus	13
2.5 Rubrik Penilaian	13
2.6 BERT.....	14
2.7 <i>Fine-tuning</i>	15

2.8 <i>Feedback</i>	15
2.9 <i>Web-based Application</i>	15
BAB III ANALISIS DAN PERANCANGAN.....	17
3.1 Analisis Sistem.....	17
3.1.1 Analisis Masalah.....	17
3.1.2 Analisis Kebutuhan.....	17
3.2 Arsitektur Umum Sistem.....	18
3.3 Pengumpulan dan pembuatan dataset.....	19
3.4 <i>Pre-Processing</i> Dataset	20
3.5 Pembangunan Model.....	21
3.6 <i>Hyperparameter Fine-tuning</i>	22
3.7 Evaluasi	22
3.8 Rubrik Penilaian dan Feedback.....	23
3.8.1 Rubrik Penilaian.....	23
3.8.2 <i>Feedback</i>	24
3.9 Perancangan Sistem.....	25
3.9.1 <i>Use Case Diagram</i>	25
3.9.2 <i>Activity Diagram</i>	26
3.9.3 <i>Flowchart</i>	26
BAB IV IMPLEMENTASI DAN PENGUJIAN	28
4.1 Implementasi Sistem	28
4.2 Implementasi <i>Website</i>	28
4.3 <i>Pre-processing</i> Dataset	34
4.4 Pengembangan Model.....	34
4.5 <i>Hyperparameter Fine-tuning</i>	37
4.6 Evaluasi	37
4.7 Rubrik Penilaian	38
4.8 <i>Feedback</i>	39
4.9 Pengujian.....	41
BAB V PENUTUP.....	44
5.1 Kesimpulan.....	44
5.2 Saran.....	44
DAFTAR PUSTAKA.....	45

DAFTAR TABEL

Tabel 3.1 Perbandingan jumlah <i>essay</i> pada setiap <i>essay_id</i>	20
Tabel 3.2 Kategori penilaian setiap parameter	24
Tabel 3.3 <i>Feedback</i> berdasarkan kategori parameter penilaian	24
Tabel 4.1 Perhitungan penilaian panjang esai	39
Tabel 4.2 Perhitungan penilaian kekayaan kosakata	39
Tabel 4.3 Pengujian pertama	41
Tabel 4.4 Pengujian kedua	42
Tabel 4.5 Pengujian ketiga	43

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Model <i>Transformers</i>	9
Gambar 2.2 <i>Pre-training</i> dan <i>Fine-tuning</i> BERT	14
Gambar 3.1 Arsitektur Umum Sistem	18
Gambar 3.2 <i>Use Case Diagram</i>	25
Gambar 3.3 <i>Activity Diagram</i>	26
Gambar 3.4 <i>Flowchart</i>	27
Gambar 4.1 <i>Landing Page</i>	29
Gambar 4.2 <i>Login Page</i>	30
Gambar 4.3 <i>Signup Page</i>	30
Gambar 4.4 <i>Dashboard</i>	31
Gambar 4.5 Tampilan Pilihan Mata Kuliah	31
Gambar 4.6 Tampilan Menu Analitik	32
Gambar 4.7 Tampilan Menu Menambah Kuis	32
Gambar 4.8 Tampilan Menu Menjawab Kuis	33
Gambar 4.9 Tampilan Hasil Kuis	33
Gambar 4.10 Tampilan Menu Pengaturan Bobot Penilaian	34
Gambar 4.11 Penghapusan Karakter Khusus	34
Gambar 4.12 Pengubahan tipe data menjadi <i>string</i>	35
Gambar 4.13 Penggabungan kolom <i>prompt</i> dan <i>essay</i>	35
Gambar 4.14 Pengskalaan nilai holistik	35
Gambar 4.15 Tokenisasi teks	35
Gambar 4.16 Lapisan tambahan <i>neural network</i>	36
Gambar 4.17 Pembuatan model	36
Gambar 4.18 Konversi <i>tensor</i> menjadi <i>array</i>	36
Gambar 4.19 Pemisahan dataset <i>training</i> dan <i>validation</i>	36
Gambar 4.20 Pengaturan <i>batch size</i> pelatihan	37
Gambar 4.21 Penggunaan metode <i>early stopping</i>	37
Gambar 4.22 Pelatihan model	37
Gambar 4.23 Hasil evaluasi dengan MSE	38
Gambar 4.24 Hasil evaluasi dengan QWK	38
Gambar 4.25 Penilaian panjang esai	38
Gambar 4.26 Penilaian kekayaan kosakata	39

Gambar 4.27 Pembagian kategori setiap parameter	40
Gambar 4.28 Pemberian <i>feedback</i>	40

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam konteks penelitian, dataset adalah sebuah hal yang krusial, terutama dalam bidang ilmu *data science* dan *machine learning*. Dataset adalah sekumpulan data yang telah diorganisir dengan baik yang dipergunakan untuk analisa serta pengembangan model. Dataset bisa berupa sekumpulan angka, gambar, teks, grafik maupun yang lainnya tergantung pada tujuan penelitian (Chapman et al., 2020). Kualitas dan kelengkapan dataset merupakan faktor penentu yang krusial dalam menentukan keberhasilan suatu penelitian, karena dengan adanya dataset yang baik dan relevan, peneliti dapat menguji hipotesis yang diangkat dalam penelitian serta menghasilkan kesimpulan yang valid.

Perkembangan zaman turut memaksa dunia pendidikan untuk mengubah sistem pembelajarannya agar dapat berjalan beriringan dengan teknologi. Salah satu contohnya yaitu menggunakan teknologi dalam mengevaluasi hasil pembelajaran. Seperti penggunaan *Automated Essay Scoring* dalam menilai soal esai. Penggunaan soal esai ataupun soal berbasis HOTS (*High Order Thinking Skill*) lainnya seperti *project based* ataupun *case method*, dikarenakan metode ini mampu menguji kemampuan analisis serta cara berpikir kritis seseorang (Qasrawi & Beniabdelrahman, 2020).

Soal esai mempunyai tantangan tersendiri bagi para pengajar, terutama ketika penilaian. Banyaknya esai yang harus dinilai dalam satu waktu bersamaan rentan terjadi kesalahan penilaian, apalagi setiap mahasiswa memiliki jawaban yang bervariasi. Hal ini menyebabkan penilaian menjadi tidak lagi objektif (Terence, 2022). Dalam proses penilaian, juga diperlukan rubrik penilaian yang mencakup berbagai aspek, seperti pemahaman materi, pengembangan ide, struktur esai dan juga kesesuaian argumen. Ada 2 tipe rubrik penilaian, yaitu rubrik holistik dan analitik. Rubrik holistik mengevaluasi sebuah karya secara keseluruhan, sedangkan rubrik analitik mengevaluasi setiap kriterianya secara terpisah (Hussein et al., 2020).

Automated Essay Scoring (AES) merupakan sebuah metode penilaian esai berbasis komputer yang mampu memeriksa hasil kerja pelajar secara otomatis

dengan komponen nilai yang sesuai dengan keinginan pengajar (Ramesh & Sanampudi, 2022). Untuk penyelesaian AES berbasis *deep learning*, telah disusun berbagai model dataset untuk proses *fine-tuning*. Proses *fine-tuning* adalah sebuah teknik *machine learning* yang melibatkan model *pre-trained* pada dataset tertentu, yang memungkinkan mereka untuk melakukan tugas tertentu (Singh et al., 2024).

Sebagai contoh dataset ASAP (*Automated Student Assessment Prize*) merupakan dataset dengan *input single text* yaitu jawaban esai dan *output* nilai skoring esai. Model ASAP menggunakan beberapa *rater* dengan jumlah *prompt* hanya 8 namun dengan variasi jawaban esai yang lengkap dengan berbagai nilai (Wang & Li, 2022). ASAP lebih diutamakan untuk nilai holistik, yaitu penilaian keseluruhan terhadap kualitas esai tanpa memisahkan aspek-aspek tertentu secara eksplisit seperti tata bahasa, argumen, atau logika. Model ini menggabungkan semua faktor tersebut dalam satu penilaian skor yang mencerminkan keseluruhan esai. Saat ini, dataset untuk analitik masih sangat terbatas terutama untuk bahasa diluar Bahasa Inggris seperti Bahasa Indonesia.

Sistem AES yang baik adalah sistem yang mampu memberikan prediksi skor analitik berdasarkan tiap parameternya, dan juga mengidentifikasi kriteria penilaian di balik setiap rubrik dengan jelas. Selain itu, sistem juga harus mampu memberikan umpan balik (*feedback*) (Kumar & Boulanger, 2021).

Penelitian tentang *Automated Essay Scoring* sudah banyak dilakukan, seperti yang dilakukan oleh (Faradhila, 2024). Penelitian ini memanfaatkan Word2Vec dalam melakukan representasi vektor, yang digunakan untuk penilaian *similarity* antara jawaban esai dengan jawaban referensi dari dosen. Penelitian ini menggunakan dataset The Hewlett Foundation: Automated Essay Scoring, yang diambil dari kompetisi Kaggle. Dataset ini berisi 3.258 dataset yang terdiri dari esai dan juga nilai yang diberikan manusia. Dataset ini kemudian diterjemahkan ke dalam Bahasa Indonesia, lalu di latih menggunakan model “bert-base-uncased” hingga mendapatkan hasil *accuracy* sempurna sebesar 1 dan nilai *kappa score* yang baik sebesar 0.82. Model kemudian digunakan untuk memprediksi nilai esai dengan *text similarity*, rubrik skor dan BERT. Kemudian model akan memberikan umpan balik sesuai dengan nilai akhir yang didapatkan.

Penelitian yang dibuat oleh (Zhong, 2024) menguji efektivitas model *pre-trained* BERT dan deBERTa dengan menambahkan lapisan *linear* dan *dropout*

untuk meningkatkan ekstraksi fitur dan pengurangan dimensi. Metrik utama evaluasinya menggunakan *Mean Square Error* (MSE). Berdasarkan penelitian yang dilakukan, didapatkan hasil berupa kedua model ini memiliki performa yang lebih bagus setelah penambahan lapisan *linear* dan *dropout*. Model deBERTa juga diketahui memiliki nilai MSE sebesar 0,2479 yang mana lebih bagus daripada kebanyakan metode RNN.

Berdasarkan hal ini, maka penelitian ini akan membangun sebuah dataset yang akan digunakan dalam sistem AES yang dibangun. Penelitian ini menggunakan *Transformer*, karena pada *Transformer* terdapat konsep *self-attention* yang mampu melakukan pembobotan berbeda untuk setiap data yang dimasukkan. Penelitian ini akan menggunakan model IndoBERT yang merupakan model BERT yang dikhususkan untuk Bahasa Indonesia. Pemilihan penggunaan model BERT dikarenakan kata yang direpresentasikan BERT lebih dinamis. Hal ini disebabkan karena BERT mempertimbangkan kata-kata disekitarnya dengan menggunakan teknik *masking*. BERT yang menggunakan teknik pendekatan *bidirectional* ini dapat memproses teks dari dua arah sehingga BERT dapat mengatasi masalah konteks jarak jauh serta dapat mengenali hubungan antar kata serta dapat menghasilkan representasi kata yang lebih kaya dan akurat (Kaliyar, 2020). Pendekatan *fine-tuning* pada BERT berfungsi untuk menyesuaikan parameter saat dilatih pada tugas tertentu, yang memungkinkan model dapat digunakan pada berbagai tugas tanpa perlu mengubah arsitektur model tersebut (Koroteev, 2021).

Penelitian ini akan menghasilkan sebuah dataset yang fungsinya untuk mendukung sistem *Automated Essay Scoring* berbahasa Indonesia. Dataset akan terdiri dari *prompt* pertanyaan, jawaban esai, nilai dari pengajar, aspek penilaian analitik, seperti relevansi jawaban dengan *prompt*, panjang esai, korelasi antar kalimat, kekayaan kosakata yang digunakan, total nilai serta umpan balik (*feedback*). Pembuatan dataset ini bertujuan untuk mengurangi kesalahan terjemahan yang dapat terjadi apabila menggunakan dataset hasil translasi dari bahasa asing ke Bahasa Indonesia, karena kesalahan sekecil apapun dapat menghilangkan makna asli dari kalimat tersebut.

1.2 Rumusan Masalah

Dataset memegang peran penting dalam penelitian, seperti dalam bidang *data science* dan *machine learning*. Dengan adanya dataset yang terstruktur, penelitian lebih mudah mencapai kesuksesan. Namun, kendala saat ini yaitu ketersediaan dataset berkualitas dalam Bahasa Indonesia masih sangat minim, khususnya untuk keperluan *Automated Essay Scoring* (AES), yang menjadi hambatan utama dalam pengembangan model penilaian otomatis yang akurat dan efektif. Walaupun AES telah banyak diterapkan di berbagai bahasa, seperti Bahasa Inggris, Prancis, dan bahasa lainnya, ketersediaan dataset untuk AES dalam Bahasa Indonesia masih sangat terbatas, sehingga menghambat kemajuan perkembangan penelitian dan aplikasi teknologi ini pada pendidikan Indonesia. Dengan demikian, dengan menciptakan dan mengembangkan dataset berbahasa Indonesia yang berkualitas, relevan, serta sesuai dengan kebutuhan penilaian esai merupakan sebuah solusi yang mendukung implementasi AES dan meningkatkan efisiensi serta objektivitas dalam evaluasi pembelajaran di Indonesia.

1.3 Batasan Masalah

Beberapa batasan masalah yang ditemukan yaitu:

1. Menggunakan metode *deep learning* yaitu IndoBERT yang merupakan model BERT khusus untuk Bahasa Indonesia.
2. Model akan dilatih menggunakan dataset yang berisi soal ujian yang sudah dinilai oleh pengajar.
3. Parameter penilaian esai yaitu relevansi jawaban dengan *prompt*, panjang esai, korelasi antar kalimat, serta kekayaan kosakata yang digunakan.
4. Setiap parameter dibagi menjadi 3 (tiga) kelas, yaitu *Good*, *Average* dan *Poor*.
5. Program dirancang dengan bahasa Python dan diimplementasikan ke dalam *website*.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membuat dataset berbahasa Indonesia yang berfungsi untuk memaksimalkan sistem *Automated Essay Scoring*. Dengan adanya sistem ini, diharapkan mampu menilai esai sesuai dengan rubrik penilaian yang sudah disepakati.

1.5 Manfaat Penelitian

Adapun manfaat yang didapatkan dari penelitian yang dilakukan adalah:

1. Memberikan pemahaman mengenai penerapan metode *deep learning* dan *natural language processing* serta teknik *fine-tuning* BERT.
2. Mendukung peneliti yang memerlukan dataset Bahasa Indonesia dalam melakukan penelitiannya.
3. Meningkatkan efisiensi serta objektivitas dalam evaluasi pembelajaran di Indonesia.
4. Memudahkan pengajar dalam melakukan penilaian esai, serta memudahkan pelajar untuk mengevaluasi hasil belajarnya dari *feedback* yang diberikan.
5. Menjadi referensi yang berguna untuk pengembangan sistem penilaian esai otomatis kedepannya.

1.6 Metodologi Penelitian

Penelitian ini menerapkan metode penelitian seperti:

1. Studi Pustaka
Pada tahap ini, peneliti mulai melakukan studi literatur untuk mencari referensi dari berbagai sumber yang membahas tentang *Automatic Essay Scoring* (AES) untuk Bahasa Indonesia dengan fitur umpan balik (*feedback*).
2. Analisis dan Perancangan Sistem
Pada tahap ini, penulis melakukan analisis mendalam untuk semua keperluan penelitian dan merancangnya dalam bentuk diagram.
3. Implementasi Sistem
Pada tahap ini, analisis, rancangan sistem, serta model AES yang telah mendapatkan hasil evaluasi yang cukup ideal diimplementasikan ke dalam sebuah *website*.
4. Pengujian Sistem
Pada tahap ini, dilakukan uji coba sistem dengan dataset yang dibangun. Tujuannya untuk mengetahui kinerja semua fitur sudah sesuai dalam melakukan penilaian esai berbahasa Indonesia dengan menggunakan dataset yang dibuat.
5. Dokumentasi Sistem
Penelitian didokumentasikan ke dalam bentuk penyusunan laporan akhir sesuai dengan format penulisan penelitian skripsi.

1.7 Penelitian Relevan

Beberapa penelitian terdahulu yang relevan dengan penelitian ini adalah sebagai berikut:

1. Penelitian oleh Mahendra et al., 2021 yang berjudul “*IndoNLI: A Natural Language Inference Dataset for Indonesian*” memperkenalkan IndoNLI, sebuah dataset Bahasa Indonesia yang terdiri dari 18.000 pasang kalimat. Hasil penelitian menunjukkan bahwa model XLM-R mengungguli model lainnya pada data IndoNLI, meskipun performa terbaik pada data yang dianotasi oleh pakar masih memiliki kesenjangan akurasi sebesar 13,4% di bawah performa manusia.
2. Penelitian oleh Koto et al., 2020 yang berjudul “*IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP*” memperkenalkan dataset IndoLEM dan model IndoBERT. IndoBERT menunjukkan kinerja terbaik dalam sebagian besar tugas dibandingkan model *pre-trained* lainnya, seperti mBERT dan MalayBERT. Untuk tugas semantik, IndoBERT unggul 13,2 poin dibandingkan Naive Bayes, dan 7,5 poin dari mBERT.
3. Penelitian oleh Reddy et al., 2019 yang berjudul “*CoQA: A Conversational Question Answering Challenge*” memperkenalkan dataset CoQA, yang berhasil mencapai *f1-score* sebesar 65,4%.
4. Penelitian oleh Rajpurkar et al., 2015 yang berjudul “*SQuAD: 100,000+ Questions for Machine Comprehension of Text*” memperkenalkan dataset Stanford Question Answering Dataset (SQuAD), berhasil mencapai *f1-score* sebesar 51%, jauh lebih baik daripada baseline sederhana, tetapi masih di bawah performa manusia yang bisa mencapai 86,8%.
5. Penelitian oleh Chen & He, 2013 yang berjudul “*Automated Essay Scoring by Maximizing Human-machine Agreement*” membagi dataset ASAP menjadi *prompt specific rating* model dan *generic rating* model. Hasil penelitian mengungkapkan bahwa pendekatan yang diajukan berhasil mencapai tingkat kesesuaian yang tinggi dengan penilaian manusia, dengan nilai sekitar 0,80 yang diukur menggunakan *quadratic weighted kappa*.

1.8 Sistematika Penulisan

Sistematika penulisan skripsi yang digunakan pada penelitian ini adalah:

BAB I PENDAHULUAN

Bab ini mencakup penjelasan mengenai latar belakang pemilihan judul, rumusan dan batasan masalah, tujuan, manfaat, metodologi penelitian, penelitian relevan, dan sistematika penulisan skripsi.

BAB II LANDASAN TEORI

Bab ini menguraikan berbagai teori yang mendukung penelitian ini, seperti *Dataset*, *Automated Essay Scoring*, *Deep Learning*, *Natural Language Processing*, *Rubrik Penilaian*, *BERT*, *Fine-tuning*, *Feedback*, *Web-based Application*.

BAB III ANALISIS DAN PERANCANGAN

Bab ini menjelaskan mengenai analisis masalah penelitian ini dan perancangan model serta diagram yang diperlukan.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi penjelasan mengenai implementasi dan pengujian sistem berdasarkan tahapan pada bab analisis dan perancangan.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang didapatkan setelah melakukan penelitian dan saran atau masukan dari peneliti untuk penelitian berikutnya.

BAB II

LANDASAN TEORI

2.1 Dataset

Dataset merupakan sekumpulan data yang telah diorganisir dengan baik yang dipergunakan untuk analisa serta pengembangan model. Dataset bisa berupa sekumpulan angka, gambar, teks, grafik maupun yang lainnya tergantung pada tujuan penelitian (Chapman et al., 2020). Umumnya, pada dataset yang berukuran sangat besar, sering kali terdapat data yang tidak cocok dengan keperluan penelitian. Inilah alasan mengapa proses penemuan dan eksplorasi data menjadi langkah yang sangat penting, walaupun memakan waktu banyak dalam alur kerja analisis data (Paton et al., 2023).

2.2 *Automated Essay Scoring*

Merupakan sebuah metode penilaian esai berbasis komputer yang mampu memeriksa hasil kerja pelajar secara otomatis dengan komponen nilai yang sesuai dengan keinginan pengajar (Ramesh & Sanampudi, 2022). Banyaknya aktivitas kegiatan kelas baik luring maupun daring di semua tingkat pendidikan di Indonesia menjadikan *Automated Essay Scoring* sebagai sebuah kebutuhan dalam menentukan uji kompetensi akademis (Buditjahjanto et al., 2022). Keefektifan sistem AES menghasilkan daya tarik kuat bagi sekolah dan universitas di dunia untuk mengimplementasikan sistem ini. Dengan begitu, pengeluaran biaya seperti honor penilai dapat dikurangi (Beseiso et al., 2021).

Automated Essay Scoring sendiri sudah diteliti sejak tahun 1967 oleh Ellis B. Page dengan menggunakan analisis regresi untuk dibandingkan dengan nilai manusia. Namun, kekurangan dari metode ini adalah model yang dihasilkan lebih berfokus pada fitur umum seperti rata-rata panjang esai dibandingkan dengan isi konten. Seiring perkembangan teknologi, (Mizumoto & Eguchi, 2023) melakukan penelitian AES dengan menggunakan GPT dan mengungkapkan bahwa penggunaan fitur linguistik dapat meningkatkan akurasi penilaian.

Automated Essay Scoring dibagi menjadi tiga pendekatan, salah satunya adalah *machine learning framework* yang menggunakan metode regresi dan klasifikasi. Pertama, dataset yang berisi kumpulan esai akan di *pre-process* untuk mengekstrak fitur esai, hingga memberikan penilaian esai sesuai dengan

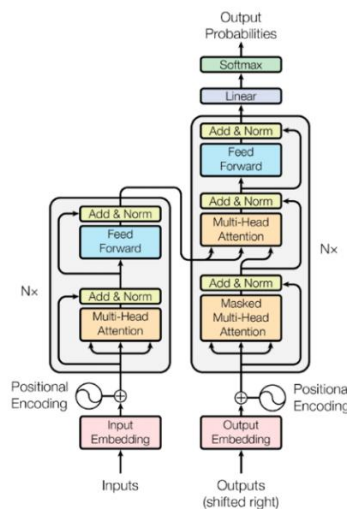
klasifikasi yang sudah ditentukan. Dalam melakukan *pre-processing*, AES melakukan pendekatan dengan teknik *Natural Language Processing*, yaitu memisah setiap kata yang ada pada esai menjadi sebuah token. Token-token ini kemudian akan dijadikan sebagai *input* untuk melatih model *machine learning*. Setelah itu, *neural networks* di lapisan tersembunyi akan melakukan tugasnya yaitu menganalisa sintaks dan semantik pada teks, hingga menghasilkan keluaran seperti nilai atau kategorinya (Uto et al., 2020).

2.3 Deep Learning

Merupakan subbagian dari *machine learning* dan *artificial intelligence* yang menitikberatkan pada pengembangan jaringan saraf yang kompleks, yang umumnya diterapkan di berbagai bidang seperti analisis teks, pengenalan visual, kesehatan, dan bidang lainnya (Sarker, 2021). *Deep learning* mampu menangani jumlah data yang sangat besar, karena *deep learning* meniru cara kerja otak manusia dalam memproses data melalui gabungan input, bobot serta bias. Hal inilah yang memungkinkan jaringan saraf untuk belajar dan mengenali pola dengan tingkat akurasi yang tinggi.

2.3.1 Transformers

Transformer pertama kali diperkenalkan oleh (Vaswani et al., 2017) pada makalah yang berjudul “Attention is All You Need”. *Transformer* dibangun menggunakan mekanisme *self-attention* dan lapisan *fully connected* untuk kedua *encoder* dan *decoder*.



Gambar 2.1 Arsitektur Model *Transformers*

(Vaswani et al., 2017)

Pada lapisan *encoder*, akan berlangsung dua mekanisme yaitu *multi-head attention* dan *feed-forward neural networks*. Pada mekanisme *multi-head attention*, input akan diubah menjadi representasi yang abstrak, dimana *encoder* akan memperhitungkan semua token dan mencari hubungan antar token tersebut. Kemudian pada proses *feed-forward*, representasi yang didapat dari blok *multi-head attention* akan diproses. Sedangkan pada lapisan *decoder*, akan diambil representasi yang dihasilkan oleh *encoder* untuk menghasilkan *input sequences*. Perbedaan utama *decoder* dengan *encoder* terdapat pada lapisan pertamanya. Pada *decoder*, tambahan teknik *masking* berfungsi untuk mencegah kesalahan pengaksesan informasi. Hal ini bertujuan untuk memastikan hasil prediksi pada *index* ke-*i* hanya akan bergantung pada keluaran sebelumnya.

Transformer sendiri sudah banyak digunakan untuk berbagai tugas dalam bidang NLP, seperti klasifikasi teks, peringkasan teks dan penerjemahan otomatis (Khan et al., 2022).

2.3.2 Hyperparameter

Hyperparameter adalah pengaturan yang diberikan sebelum proses *training* model. *Hyperparameter* bertujuan untuk mendapatkan kombinasi antar pengaturan agar model yang dihasilkan bisa mendapatkan hasil yang paling optimal. *Hyperparameter* dibagi menjadi tiga jenis, yaitu *continuous hyperparameters*, *discrete hyperparameters* dan *categorical hyperparameters* (Yang & Shami, 2020). Adapun *hyperparameters* yang umum digunakan pada model *machine learning* yaitu *learning rate*, *batch size* serta *activation function*.

2.3.3 Learning Rate

Learning rate berfungsi untuk memperbarui *weights* / bobot selama proses pelatihan. Nilai dari *learning rate* harus optimal. Jika nilai *learning rate* terlalu tinggi, algoritma optimasi akan melewatkan informasi penting yang mengakibatkan model menjadi tidak mendapatkan pembelajaran yang seharusnya dipelajari oleh model. Namun, jika nilai *learning rate* terlalu rendah, proses pelatihan akan berlangsung lama dan sulit mencapai konvergen. Oleh karena itu, penentuan nilai *learning rate* yang

tepat merupakan langkah yang sangat penting agar model dapat stabil selama proses pelatihan.

2.3.4 *Batch Size*

Merupakan jumlah data atau sampel yang diproses oleh model dalam satu langkah sebelum bobotnya diperbarui. Ukuran *batch size* memengaruhi kecepatan dan stabilitas pelatihan. *Batch size* yang kecil lebih sering memperbarui bobotnya. Sementara, *batch size* yang besar menghasilkan pembaruan yang lebih stabil, namun memerlukan lebih banyak memori pada komputer.

Sebagai contoh, jika dataset memiliki 10.000 data dan *batch size* adalah 16, maka data akan dibagi menjadi 625 *batch* dengan setiap *batch* terdiri dari 16 data. Dengan kata lain, model akan memproses 16 data secara bersamaan sebelum ia memperbarui bobotnya.

2.3.5 *Activation Function*

Fungsi aktivasi (*activation function*) adalah komponen pada *neural networks* yang memungkinkan jaringan untuk memahami pola serta hubungan yang kompleks pada data. Fungsi ini bekerja dengan memproses hasil penjumlahan bobot *input* dan bias di setiap *neuron* dan mengubahnya melalui proses matematis tertentu, hingga menghasilkan keluaran yang akan digunakan oleh lapisan berikutnya.

Jenis-jenis fungsi aktivasi adalah sebagai berikut:

1. Fungsi Sigmoid

Fungsi sigmoid memiliki nilai yang berkisar pada skala 0 hingga 1. Sigmoid sering digunakan di lapisan *output* untuk kasus klasifikasi biner. Namun, kelemahan utama sigmoid adalah *vanishing gradient*, di mana gradien menjadi sangat kecil untuk nilai *input* besar atau kecil, yang menghambat pembaruan bobot selama pelatihan.

Formula:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

2. Fungsi Tanh (*Hyperbolic Tangent*)

Fungsi tanh memetakan *output* antara nilai -1 hingga 1. Fungsi tanh merupakan penguatan dari fungsi sigmoid dan bersifat simetris terhadap nol, yang membuatnya lebih cocok digunakan untuk data yang memiliki nilai positif dan negatif. Namun, tanh juga memiliki kekurangan yaitu *vanishing gradient*.

Formula:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

3. Fungsi ReLU (*Rectified Linear Unit*)

Fungsi ReLU banyak digunakan karena kesederhanaannya dan efisiensinya dalam menghindari *vanishing gradient*. ReLU hanya mengaktifkan *neuron* untuk nilai *input* positif, sehingga membantu jaringan dalam pembelajaran lebih cepat dan stabil. Kelemahan ReLU adalah *dead neuron problem*, dimana beberapa *neuron* menjadi tidak aktif karena terjebak di nilai 0.

Formula:

$$f(x) = \max(0, x) \quad (3)$$

2.4 Natural Language Processing

Natural Language Processing (NLP) merupakan bidang ilmu *artificial intelligence* (AI) yang memfasilitasi bagaimana komputer dapat berinteraksi dengan manusia menggunakan bahasa yang kita pahami. Teknologi ini memungkinkan komputer untuk memahami bahasa yang digunakan manusia dan memberikan balasan baik dalam bentuk teks ataupun suara. NLP dibagi menjadi dua, yaitu NLU (*Natural Language Understanding*) yang berperan dalam memberikan kemampuan kepada mesin untuk memahami serta menganalisis bahasa alami, termasuk menangkap makna, konteks dan struktur dari teks dan NLG (*Natural Language Generation*) merupakan proses menghasilkan frasa, kalimat, atau paragraf yang koheren dan bermakna berdasarkan representasi internal data, sehingga dapat disampaikan kembali dalam bentuk yang mudah dipahami oleh manusia (Khurana et al., 2023).

2.4.1 Tokenisasi

Proses pemecahan teks menjadi unit yang lebih kecil, berupa kata ataupun karakter disebut tokenisasi. Dengan adanya tokenisasi, makna dari setiap token lebih mudah terekstraksi dan kata-kata baru atau yang jarang muncul juga lebih efisien ditangani. Hal ini membuat proses dalam NLP menjadi lebih efektif.

2.4.2 Penghapusan Karakter Khusus

Penghapusan karakter khusus adalah proses NLP dimana teks yang mengandung elemen-elemen tidak diperlukan, seperti simbol, tanda baca, angka, atau karakter non-alfanumerik akan dihapus. Langkah ini bertujuan untuk meningkatkan kualitas teks dengan menghilangkan elemen yang dapat mengganggu proses analisis. Metode yang paling banyak digunakan yaitu penggunaan *regular expressions* (regex). Teknik ini memungkinkan penghapusan karakter tertentu seperti spasi yang berlebihan, tab, atau karakter garis baru (*newline*), sehingga teks yang dihasilkan lebih bersih dan terstruktur.

2.5 Rubrik Penilaian

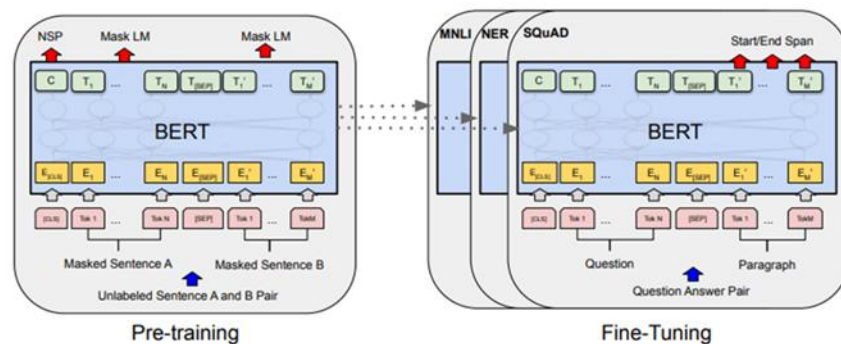
Rubrik penilaian merupakan sebuah standar ketentuan dalam menilai dan mengevaluasi kualitas kinerja seseorang. Dalam konteks ujian, rubrik penilaian mencakup berbagai kriteria seperti hasil belajar, kemampuan, maupun yang lainnya. Setiap kriteria akan diberikan tingkatan penilaian masing-masing (Hambali et al., 2022). Ada 2 tipe rubrik penilaian, yaitu rubrik holistik dan analitik. Rubrik holistik mengevaluasi sebuah karya secara keseluruhan, sedangkan rubrik analitik memecah penilaian ke dalam beberapa kriteria dan mengevaluasi setiap kriterianya secara terpisah (Hussein et al., 2020). Setiap kriteria akan mendapatkan nilainya masing-masing dan akan diberikan *feedback* yang menjelaskan performa kriterianya.

Dalam rubrik penilaian analitik, fitur penilaian dikelompokkan menjadi dua komponen utama seperti sintaksis dan semantik. Fitur sintaksis merepresentasikan aspek teknis penulisan yang mencakup struktur dan bentuk bahasa. Komponen dalam sintaksis mencakup parameter seperti panjang esai, yang mencerminkan kelengkapan isi, dan kekayaan kosakata, yang menggambarkan variasi serta

keunikan penggunaan kata. Sementara itu, fitur semantik berfokus pada makna dan keselarasan isi. Komponen semantik meliputi relevansi antara jawaban dengan *prompt*, yang menunjukkan sejauh mana jawaban memenuhi topik yang diminta, serta korelasi antar kalimat dalam esai, yang mencerminkan keterpaduan dan alur logis penulisan (Uto et al., 2020).

2.6 BERT

BERT (*Bidirectional Representations from Transformers*) adalah sebuah model *machine learning* yang berguna untuk memproses bahasa alami secara dua arah. Pada dasarnya, BERT terdiri atas beberapa lapisan *encoder Transformer* yang masing-masing memiliki beberapa 'head' dalam mekanisme *self-attention*. Setiap 'head' ini menghasilkan tiga vektor berbeda untuk setiap token input: kunci (*key*), nilai (*value*), dan *query*. Ketiga vektor tersebut digunakan untuk menghitung bobot dan membentuk representasi yang lebih kaya dari teks input. Hasil keluaran dari semua 'head' di lapisan yang sama digabungkan menjadi satu dan kemudian diproses melalui lapisan *neural network* yang terhubung sepenuhnya. Setiap lapisan juga dilengkapi dengan koneksi lanjutan untuk menjaga aliran informasi, dan diikuti dengan proses normalisasi. Seperti yang dijelaskan oleh (Rogers et al., 2020) alur kerja utama BERT dibagi menjadi dua tahap, yaitu *pre-training* dan *fine-tuning*.



Gambar 2.2 *Pre-training dan Fine-tuning BERT*

(Vaswani et al., 2017)

Pada tahap *pre-training*, model dilatih dengan menggunakan data tanpa label dan menjalani dua tugas utama, *Next Sentence Prediction* serta *Masked Language Modelling*. BERT mengonversi teks input menjadi token *embeddings*, *segment embeddings*, dan *positional encodings* (Djoko et al., 2020).

Proses ini dilakukan untuk menciptakan representasi yang bisa menangkap makna dari teks yang diberikan. Setelah tahap *pre-training* selesai, lapisan *output* ditambahkan untuk memungkinkan model menghasilkan jawaban atas pertanyaan yang diberikan.

2.7 *Fine-tuning*

Fine-tuning adalah sebuah teknik *machine learning* yang melibatkan model *pre-trained* pada dataset tertentu, yang memungkinkan mereka untuk melakukan tugas tertentu (Singh et al., 2024). Dalam pembuatan AES ini, setelah melewati tahap *pre-training*, model BERT diinisialisasi ulang pada tahapan *fine-tuning* untuk menyesuaikan dengan data berlabel dari tugas spesifik yang diinginkan. Pada tahap ini, parameter disesuaikan secara *end-to-end*, untuk menghasilkan model yang efisien untuk beragam tugas NLP (Devlin et al., 2018).

2.8 *Feedback*

Dalam konteks pendidikan, umpan balik atau *feedback* adalah pemberian tanggapan atas hasil kerja pelajar oleh pengajar (Noviani et al., 2019). Tanggapan yang diberikan dapat berupa komentar, pujian maupun saran sesuai dengan hasil kinerja mereka. Dampak dari pemberian umpan balik sangatlah fantastis, sebab berpotensi untuk meningkatkan motivasi belajar mereka. Selain itu, pemberian umpan balik juga dapat mempengaruhi *self-efficacy*.

Self-efficacy adalah rasa percaya diri seseorang yang muncul saat mereka menghadapi situasi tertentu. Penelitian menunjukkan bahwa pemberian nilai saja pada hasil evaluasi ujian dapat menurunkan tingkat *self-efficacy* mahasiswa. Sebaliknya, dengan adanya umpan balik yang bersifat deskriptif mampu meningkatkan *self-efficacy* mereka.

2.9 *Web-based Application*

Web-based application) atau yang lebih dikenal dengan aplikasi berbasis web merupakan sebuah *platform* yang mudah digunakan dan diimplementasikan dengan model *machine learning*. Saat proses *deployment* model ke dalam *website*, dibutuhkan sebuah layanan *cloud computing* yang dapat menyimpan model *machine learning* tersebut.

Ada 3 jenis layanan penyedia berbasis *cloud*, yaitu SaaS, PaaS serta IaaS (Nadeem, 2022). SaaS (*Software as a Service*) menyediakan layanan *software*

secara *online* kepada pengguna tanpa harus memedulikan aspek infrastruktur, manajemen server dan lainnya. Contohnya seperti Office 365. Pada PaaS (*Platform as a Service*), penyedia layanan memberikan layanan pengembangan dan pengujian aplikasi yang dibangun oleh pengguna tanpa perlu memikirkan pengelolaan infrastruktur ataupun sistem operasi (Yathiraju, 2022). Contoh PaaS adalah Google Cloud Platform (GCP). Sedangkan IaaS (*Infrastructure as a Service*) menyediakan layanan infrastruktur dasar seperti server, jaringan yang membuat pengguna lebih fleksibel dalam mengatur dan mengelola apa saja yang akan digunakan. Contohnya adalah Microsoft Azure Virtual Machines.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Tahapan ini bertujuan untuk mengetahui kebutuhan yang diperlukan oleh sistem untuk dapat beroperasi dengan optimal. Ada dua tahapan analisis yang digunakan, yaitu analisis masalah dan kebutuhan.

3.1.1 Analisis Masalah

Pada tahap ini, fokus permasalahan yang dikaji adalah kurangnya dataset berbahasa Indonesia untuk pengembangan sistem AES. Kurangnya dataset berbahasa Indonesia tentunya menjadi masalah yang besar, mengingat bahasa Indonesia mengandung beragam makna, klausa, serta gaya bahasa yang berbeda. Makna-makna ini tentunya akan hilang bila dataset didapatkan melalui hasil translasi dari bahasa asing. Selain itu, penilaian yang dilakukan pada umumnya adalah secara manual serta penilaian yang diberikan adalah penilaian holistik. Implementasi penerapan model *fine-tuning* IndoBERT dengan dataset bahasa Indonesia yang dibentuk dari awal menawarkan solusi untuk mengatasi permasalahan ini.

3.1.2 Analisis Kebutuhan

Masalah diatas dapat diselesaikan dengan dipenuhinya kebutuhan oleh sistem yang dirancang. Kebutuhan tersebut ialah kebutuhan fungsional dan non-fungsional.

1. Kebutuhan fungsional

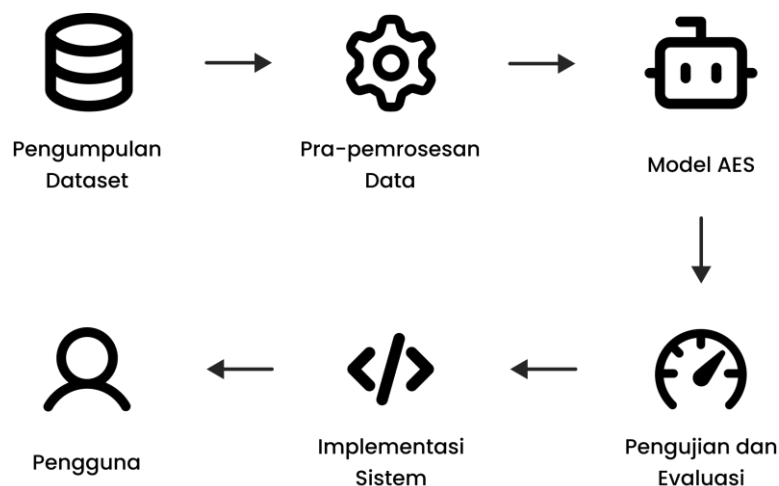
Merupakan fitur utama dari sistem yang harus diimplementasikan untuk memenuhi kebutuhan pengguna. Kebutuhan fungsional sistem mencakup beberapa hal berikut:

- a. Menerima soal dan jawaban (*prompt* dan *essay*) berbahasa Indonesia sebagai inputan.
- b. Melakukan proses *pre-processing* dataset.
- c. Menggunakan BERT untuk membangun model regresi untuk memprediksi nilai relevansi dan nilai korelasi dari esai.
- d. Memberikan skor sesuai rubrik penilaian untuk setiap esai.

- e. Nilai akhir esai didapatkan dari penjumlahan semua rubrik penilaian dikali dengan bobotnya masing-masing.
 - f. Umpan balik diberikan berdasarkan kategori yang didapatkan dari semua parameter penilaian.
 - g. Keluaran dari sistem berupa *relevancy_score*, *correlation_score*, *length_score*, *vocabulary_score*, *total_ascore* serta *feedback*.
2. Kebutuhan non-fungsional
- Merupakan fitur tambahan yang mendukung kemampuan sistem bekerja. Beberapa kebutuhan tersebut yaitu:
- a. Sistem menerapkan *authorization*, dimana sistem bisa diakses melalui sisi pelajar, pengajar dan admin.
 - b. Sistem memiliki tampilan yang *user friendly*.
 - c. Pengajar dapat menyesuaikan bobot setiap parameter penilaian.
 - d. Sistem dapat menerima inputan, memproses data dalam sekejap dan menghasilkan penilaian yang akurat.

3.2 Arsitektur Umum Sistem

Arsitektur umum sistem merupakan gambaran proses yang akan dilakukan untuk membangun keseluruhan sistem. Adapun tahapannya dapat dilihat pada gambar berikut:



Gambar 3.1 Arsitektur Umum Sistem

Langkah awal dimulai dengan mengumpulkan dataset yang akan digunakan untuk pelatihan. Dataset kemudian di *pre-process* untuk dibersihkan dan

disesuaikan dengan kebutuhan model. Selanjutnya, dilakukan pelatihan model regresi menggunakan model IndoBERT dengan *hyperparameters* yang telah ditetapkan untuk memprediksi nilai relevansi dan nilai korelasi. Setelah model dilatih, kinerja AES dievaluasi dan diuji untuk memastikan keandalannya. Setelah tidak ada lagi perbaikan, model kemudian diekspor dan diimplementasikan ke dalam API dan dihubungkan dengan *website* agar aplikasi dapat diakses dan digunakan oleh pengguna.

3.3 Pengumpulan dan pembuatan dataset

Penelitian ini menggunakan dataset yang didapatkan dari unit organisasi IKLC (Ilmu Komputer Laboratory Center), yang berisi kumpulan soal ujian praktikum yang telah dirancang oleh asisten laboratorium. Para praktikan diwajibkan untuk menjawab soal-soal tersebut, yang kemudian hasil jawabannya dinilai oleh tiga orang asisten laboratorium secara holistik. Nilai-nilai yang diberikan oleh ketiga penilai ini selanjutnya dirata-ratakan untuk menghasilkan *total_hscore*. Skor ini digunakan sebagai referensi utama dalam analisis dan pengembangan model.

Dataset ini terdiri dari 10.000 pasangan soal dan esai yang memiliki struktur sebagai berikut:

1. *essay_id* : merupakan id unik untuk setiap *prompt*
2. *prompt* : merupakan soal
3. *essay* : merupakan jawaban
4. *human_score1* : merupakan nilai yang diberikan oleh pengajar 1
5. *human_score2* : merupakan nilai yang diberikan oleh pengajar 2
6. *human_score3* : merupakan nilai yang diberikan oleh pengajar 3
7. *total_hscore* : merupakan rata-rata nilai yang diberikan ketiga pengajar (nilai holistik)
8. *relevancy_score* : merupakan nilai seberapa relevan antara *prompt* dan *essay*
9. *correlation_score* : merupakan nilai dari korelasi antar kalimat yang ada pada *essay*
10. *length_score* : merupakan nilai dari panjang *essay*
11. *vocabulary_score* : merupakan nilai dari banyaknya kosakata yang unik

12. *total_ascore* : merupakan total nilai dari setiap parameter penilaian analitik yang dikali dengan bobot masing-masing
13. *feedback* : merupakan umpan balik yang didapatkan berdasarkan kelas setiap parameter penilaian analitik

Dataset ini terdiri dari 72 *essay_id* yang masing-masing id memiliki jumlah esai yang berbeda. Berikut adalah tabel yang berisi jumlah *essay* pada masing-masing *essay_id*.

Tabel 3.1 Perbandingan jumlah *essay* pada setiap *essay_id*

<i>essay_id</i>	jumlah	<i>essay_id</i>	jumlah	<i>essay_id</i>	jumlah	<i>essay_id</i>	jumlah
1	80	19	148	37	145	55	130
2	80	20	148	38	145	56	140
3	130	21	148	39	145	57	140
4	130	22	148	40	145	58	140
5	130	23	148	41	145	59	140
6	145	24	148	42	145	60	140
7	145	25	148	43	145	61	140
8	145	26	155	44	145	62	140
9	145	27	155	45	145	63	140
10	145	28	155	46	130	64	140
11	145	29	155	47	130	65	140
12	145	30	155	48	130	66	120
13	145	31	155	49	130	67	120
14	145	32	155	50	130	68	120
15	145	33	155	51	130	69	120
16	148	34	155	52	130	70	120
17	148	35	155	53	130	71	100
18	148	36	145	54	130	72	120

3.4 Pre-Processing Dataset

Pre-processing dataset adalah rangkaian prosedur yang dilakukan untuk membersihkan dan membuang data yang tidak relevan agar dataset yang digunakan benar-benar sesuai dengan kebutuhan pelatihan. Dalam penelitian ini, dilakukan penghapusan karakter khusus yang terdapat pada jawaban esai. Proses ini bertujuan untuk menghapus karakter non-ASCII, jarak spasi yang berlebihan dalam jawaban esai dengan menggunakan *library regular expression* (regex).

3.5 Pembangunan Model

Setelah melalui tahap *pre-processing* dataset, dataset yang sudah bersih akan digunakan untuk keperluan analisis. Dalam hal ini, data yang akan dianalisis terdiri dari dua kolom utama, yaitu *prompt* dan *essay*. Kolom *prompt* berisi berbagai pertanyaan esai dan kolom *essay* memuat jawaban-jawaban atas *prompt* tersebut. Untuk memastikan kompatibilitas data, tipe data kedua kolom ini diubah menjadi tipe *string*. Setelah memastikan kedua kolom memiliki tipe data yang sesuai, langkah berikutnya adalah menggabungkan kolom *prompt* dan *essay* ke dalam satu kolom baru. Penggabungan ini dilakukan dengan menggunakan *string separator* [SEP]. [SEP] merupakan token khusus yang sering digunakan dalam model BERT untuk memisahkan dua bagian teks dalam satu *input*.

Tahap selanjutnya adalah melakukan normalisasi nilai *total_hscore*. Proses ini berfungsi untuk mengubah nilai ke dalam rentang 0 dan 1. Tujuannya agar semua nilai memiliki skala yang seragam. Dengan begini, model akan lebih mudah mempelajari dan memahami pola data. Nilai yang sudah di normalisasi kemudian digunakan untuk menghitung dua metrik penilaian utama, yaitu *relevancy_score* dan *correlation_score*.

Langkah berikutnya yaitu memproses data tersebut menggunakan *tokenizer* IndoBERT. *Tokenizer* yang digunakan pada penelitian ini adalah model *pre-trained* indobenchmark/indobert-base-p2 yang dikembangkan khusus untuk Bahasa Indonesia. *Tokenizer* kemudian memecah pertanyaan dan jawaban yang sudah digabung menjadi representasi token yang siap digunakan oleh model. Setelah data berhasil ditokenisasi, representasi vektor dari teks esai dihitung menggunakan model IndoBERT. Model ini memanfaatkan arsitektur *Transformer* dengan mekanisme *self-attention* untuk menangkap hubungan antar kata dalam konteks yang lebih luas. Model kemudian akan dilatih untuk memahami pola dan hubungan antara teks esai dan soal, agar dapat memprediksi skor relevansi dan korelasi secara akurat. Proses pelatihan akan diatur dengan beberapa *hyperparameter* penting seperti *learning rate*, *batch size*, *epoch* dan lainnya. Terakhir, model akan dievaluasi menggunakan dua buah metrik evaluasi yaitu *Mean Squared Error* (MSE) dan *Quadratic Weighted Kappa* (QWK).

3.6 Hyperparameter Fine-tuning

Penentuan *hyperparameter* dalam pengembangan model adalah hal yang krusial, karena *hyperparameter* berperan langsung dalam mengontrol kinerja dan stabilitas pelatihan model. Pemilihan *hyperparameter* seperti *learning rate*, *batch size*, dan jumlah *epoch* dapat memengaruhi seberapa cepat model belajar, sejauh mana model dapat menangkap pola dalam data, serta kemampuan model untuk menghindari *overfitting* atau *underfitting*.

Learning rate yang terlalu tinggi dapat menyebabkan model gagal mencapai konvergensi karena pembaruan bobot yang terlalu agresif, sementara *learning rate* yang terlalu rendah dapat memperlambat proses pelatihan dan membuat model sulit menemukan solusi optimal. *Batch size* yang besar dapat mempercepat pelatihan dengan memanfaatkan paralelisme pada perangkat keras, namun memerlukan memori yang lebih besar. Sebaliknya, *batch size* yang kecil memungkinkan model untuk mempelajari hal yang lebih rinci dan detail, tetapi pelatihan akan memakan waktu yang lebih lama. Jumlah *epoch* juga sangat berpengaruh, di mana jumlah *epoch* yang terlalu kecil mungkin membuat model gagal memahami pola yang kompleks, sementara jumlah *epoch* yang terlalu besar berisiko menyebabkan model menjadi *overfitting*.

Dengan pengaturan *hyperparameter* yang tepat, model dapat mencapai kinerja optimal dalam memprediksi *relevancy_score* dan *correlation_score* dengan akurasi yang tinggi.

3.7 Evaluasi

Evaluasi dilakukan dengan menggunakan MSE (*Mean Squared Error*) dan QWK (*Quadratic Weighted Kappa*). Kedua metrik ini dapat memberikan informasi yang jelas tentang seberapa baik model memprediksi nilai kontinu, seperti *relevancy_score* dan *correlation_score*. MSE akan menghitung rata-rata dari kuadrat selisih antara nilai hasil prediksi dengan nilai aktual, sehingga memberikan penalti yang lebih besar terhadap kesalahan yang lebih besar. Sedangkan, QWK mengukur kesesuaian antara nilai hasil prediksi dengan nilai sebenarnya dalam skala ordinal. QWK akan memberikan bobot yang lebih besar pada kesalahan yang lebih jauh dari nilai yang sebenarnya, sehingga memberikan evaluasi yang lebih akurat terhadap kemampuan model dalam memprediksi nilai secara konsisten.

3.8 Rubrik Penilaian dan *Feedback*

3.8.1 Rubrik Penilaian

Rubrik penilaian dibutuhkan untuk memberikan standar yang jelas dan objektif dalam mengevaluasi kualitas esai. Dengan adanya rubrik penilaian, setiap elemen esai seperti relevansi jawaban dengan *prompt*, korelasi antar kalimat, panjang esai, dan kekayaan kosakata, dapat dinilai secara terpisah namun saling mendukung untuk memberikan gambaran yang komprehensif mengenai kualitas keseluruhan esai. Rubrik juga dapat membantu mengurangi subjektivitas dalam penilaian serta memberikan umpan balik kepada penulis esai.

1. Relevansi jawaban dengan *prompt*

Hal ini mengukur seberapa relevan jawaban yang ditulis dengan soal yang diberikan. Secara *default*, bobot untuk penilaian ini adalah 0.25.

2. Korelasi antar kalimat

Hal ini mengukur seberapa padu dan logis alur penulisan esai. Secara *default*, bobot untuk penilaian ini adalah 0.45.

3. Panjang esai

Hal ini mengukur jumlah kata dalam sebuah esai. Secara *default*, bobot untuk penilaian ini adalah 0.15. Kalkulasi penilaiannya yaitu:

- Jika jumlah kata lebih dari atau sama dengan 200 kata, maka nilai yang didapatkan adalah 100.
- Jika jumlah kata lebih dari atau sama dengan 100 tapi kurang dari 200, maka nilai akan dihitung berdasarkan rumus fungsi eksponensial berikut:

$$50 + 40 \times (1 - e^{-0.02 \times (\text{word_count} - 100)}) \quad (4)$$

- Jika jumlah kata kurang dari 100, maka nilai akan dihitung berdasarkan rumus berikut:

$$25 + 25 \times (1 - e^{-0.04 \times \text{word_count}}) \quad (5)$$

4. Kekayaan kosakata

Hal ini mengukur jumlah kata unik yang ada dalam esai. Nilai dihitung menggunakan *Type-Token Ratio* (TTR) yang merupakan rasio antara

jumlah kata unik dengan total kata pada esai. Secara *default*, bobot untuk penilaian ini adalah 0.15.

3.8.2 *Feedback*

Umpan balik (*feedback*) diberikan berdasarkan kategori kelas yang diperoleh masing-masing parameter penilaian analitik. Setiap parameter penilaian dibagi menjadi 3 kategori yaitu *Good*, *Average* dan *Poor*. Semakin banyak nilai yang mendapatkan kategori *Good*, artinya esai yang dibuat semakin berkualitas. Sebaliknya, jika sebagian besar nilai mendapatkan kategori *Poor*, maka esai dianggap kurang memadai dan membutuhkan perbaikan yang signifikan.

Berikut ini adalah pengelompokan kelas untuk setiap parameter penilaian.

Tabel 3.2 Kategori penilaian setiap parameter

Kelas	Nilai Relevansi	Nilai Korelasi	Nilai Panjang Esai	Nilai Kosakata
Good	>70	>80	>90	>70
Average	40 – 70	50 – 80	66 – 90	50 – 70
Poor	<40	<50	<66	<50

Adapun kombinasi dari setiap kelas yang didapat oleh parameter penilaian untuk menghasilkan *feedback* adalah sebagai berikut.

Tabel 3.3 *Feedback* berdasarkan kategori parameter penilaian

<i>Good</i>	<i>Average</i>	<i>Poor</i>	<i>Feedback</i> yang diberikan
4	0	0	Kualitas esai luar biasa, memenuhi ketentuan di semua aspek.
3	1	0	Kualitas esai sangat baik dengan sedikit aspek yang perlu disempurnakan.
3	0	1	Kualitas esai yang bagus, meskipun ada beberapa aspek yang bisa ditingkatkan lagi.
2	2	0	Secara keseluruhan kualitas esai baik, namun masih terdapat ruang untuk pengembangan.
2	1	1	Kualitas esai cukup baik, meskipun beberapa aspek mengurangi kualitasnya secara keseluruhan.
2	0	2	Kualitas esai masih dapat diterima, namun memerlukan perbaikan khusus.

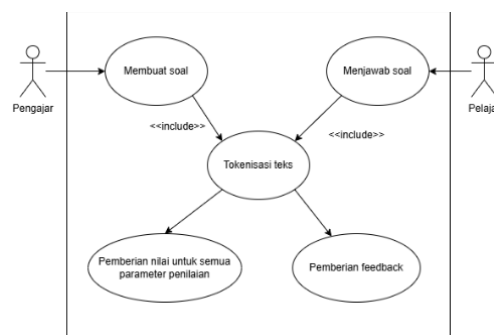
1	3	0	Kualitas esai sedang, akan lebih baik dengan peningkatan di berbagai bidang.
1	2	1	Kualitas esai terbatas karena beberapa aspek pada esai sangat terbatas.
1	1	2	Kualitas esai kurang memadai, beberapa bagian membutuhkan perbaikan signifikan.
1	0	3	Kualitas esai tidak memadai, masih terdapat banyak kekurangan.
0	4	0	Kualitas esai pada tingkat biasa, konsisten pada tingkat rata-rata.
0	3	1	Kualitas esai cukup memadai, namun ada kelemahan yang bisa diperbaiki.
0	2	2	Kualitas esai tidak memuaskan, masih kurang pada aspek-aspek penting.
0	1	3	Kualitas esai sangat buruk, perlu banyak perbaikan signifikan untuk menghasilkan esai yang bagus.
0	0	4	Esai tidak dapat diterima, kualitasnya jauh dari kata sempurna karena gagal memenuhi standar dasar pada semua aspek.

3.9 Perancangan Sistem

Dalam merancang sistem yang diinginkan, diperlukan sejumlah diagram yang dapat menjelaskan spesifikasi sistem secara rinci. Selain itu, perlu diperhatikan alur proses dalam tahap perancangan program untuk memastikan adanya konsistensi dan keselarasan yang sesuai.

3.9.1 Use Case Diagram

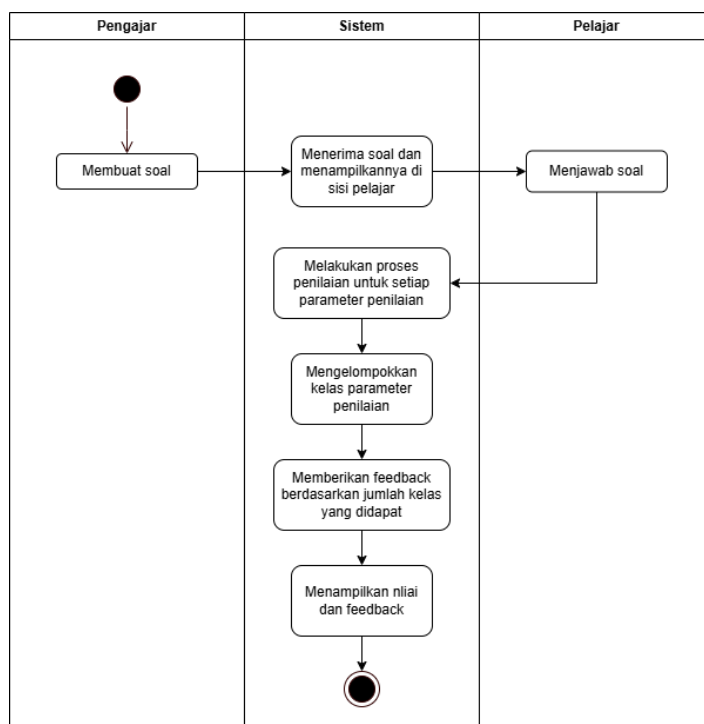
Merupakan diagram yang memvisualisasikan interaksi pengguna dengan sistem yang dirancang. *Use case* akan menampilkan fitur utama yang dimiliki sistem, serta hubungannya dengan aktor (pengguna). Pengimplementasian *use case* membantu pengguna mengetahui kebutuhan dasar sebuah sistem, sehingga berguna dalam tahap analisis dan perancangan sistem.



Gambar 3.2 Use Case Diagram

3.9.2 Activity Diagram

Activity Diagram biasanya digunakan untuk memodelkan alur kerja atau proses bisnis, baik dalam sistem maupun interaksi antara *user* dengan sistem. Diagram ini merepresentasikan aktivitas-aktivitas yang saling berhubungan, termasuk kondisi serta urutan dalam proses tersebut. *Activity Diagram* memberikan visualisasi yang jelas mengenai bagaimana proses berlangsung dari awal hingga akhir. Diagram ini membantu dalam analisis, perancangan, dan pengoptimalan proses, sehingga sangat berguna dalam pengembangan perangkat lunak dan pemodelan bisnis.



Gambar 3.3 Activity Diagram

3.9.3 Flowchart

Flowchart adalah ilustrasi grafis yang merepresentasikan proses, alur kerja, atau algoritma yang menggambarkan langkah-langkah secara berurutan menggunakan simbol. Setiap simbol memiliki fungsi tertentu. *Flowchart* dapat mempermudah pemahaman, analisis, dan komunikasi mengenai bagaimana suatu proses atau sistem bekerja. Dengan visualisasi yang jelas, *flowchart* membantu mengidentifikasi hambatan, mengoptimalkan proses, dan memastikan bahwa setiap langkah dirancang secara logis.



Gambar 3.4 *Flowchart*

BAB IV

IMPLEMENTASI DAN PENGUJIAN

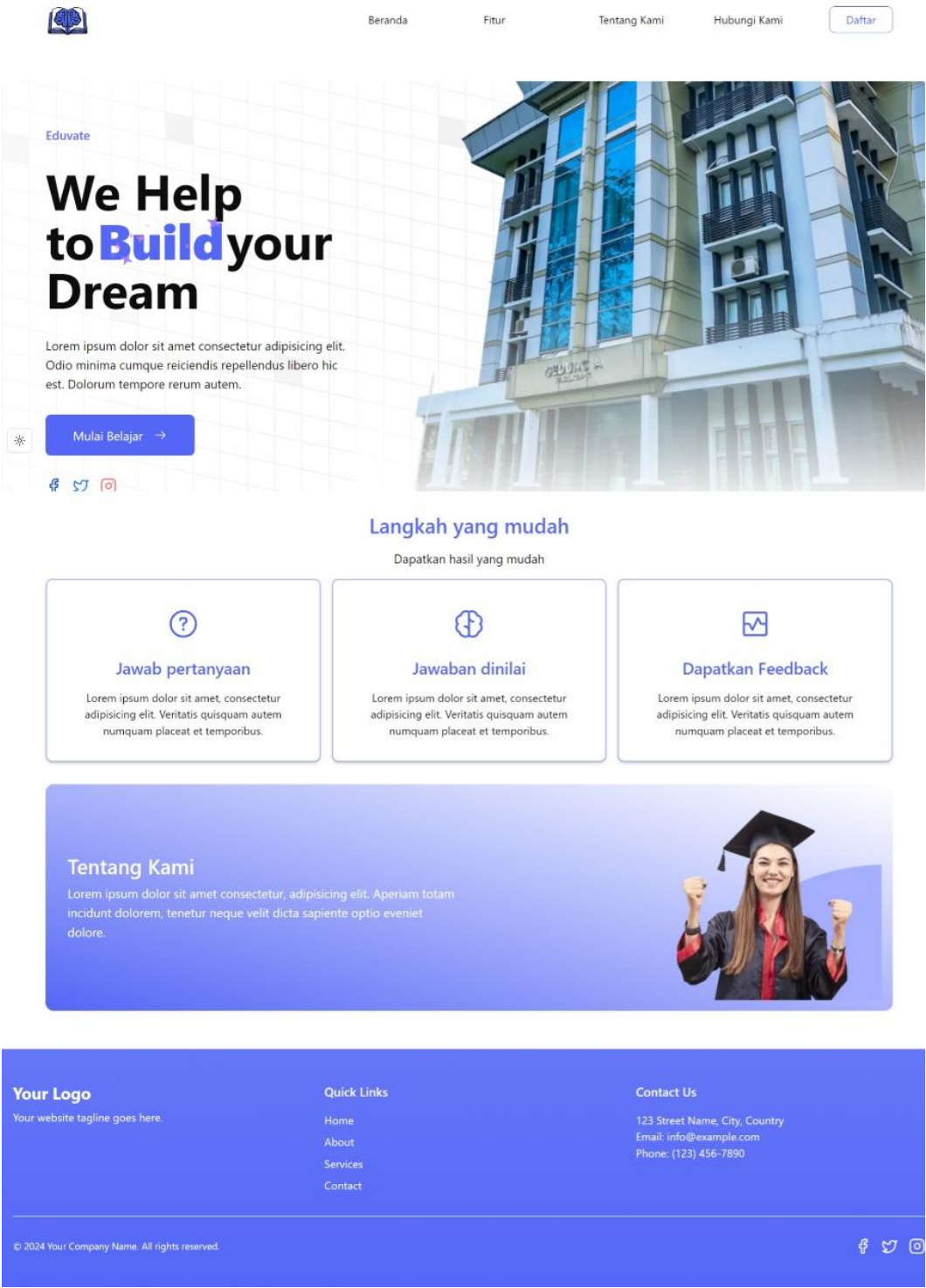
4.1 Implementasi Sistem

Sistem dikembangkan secara terstruktur sesuai dengan perancangan yang telah dijelaskan sebelumnya, dengan memanfaatkan perangkat keras dan lunak di bawah ini:

1. Processor Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz
2. RAM 8GB
3. Sistem operasi Windows 11 Home Single Language 64-bit
4. Browser Google Chrome
5. Google Colaboratory Pro
6. Text Editor Visual Studio Code
7. Python 3.9.12
8. Tensorflow 2.9.1
9. Transformers 4.21.1
10. Regex 2024.9.11
11. Pandas 2.2.3
12. Numpy 1.23.0
13. Scikit-learn 1.5.2
14. Flask 3.1.0
15. Flask-Cors 5.0.0
16. Next.js
17. MongoDB

4.2 Implementasi Website

Berikut ini adalah hasil implementasi sistem pada *website*. *Website* ini dibangun menggunakan *framework* Next.js untuk mendesain dan mengatur tampilan utama, memastikan pengalaman pengguna yang cepat dan responsif. Sebagai penyimpanan data, digunakan MongoDB yang dikenal dengan skalabilitasnya dan kemampuannya menangani data dalam jumlah besar. Serta untuk menghubungkan *website* dengan model yang telah dilatih sebelumnya, digunakan REST API, yang memungkinkan komunikasi data secara efektif antara *frontend* dan *backend*.



Gambar 4.1 Landing Page

Masuk

Daftar

Masukkan informasi Anda untuk mengakses akun Anda

Email

m@example.com

Password

Login

Belum punya akun? [Daftar](#)

Kembali ke Beranda



Gambar 4.2 Login Page

Masuk

Daftar

Buat akun baru Anda

Name

John Doe

Email

m@example.com

Kata sandi

Ulangi kata sandi

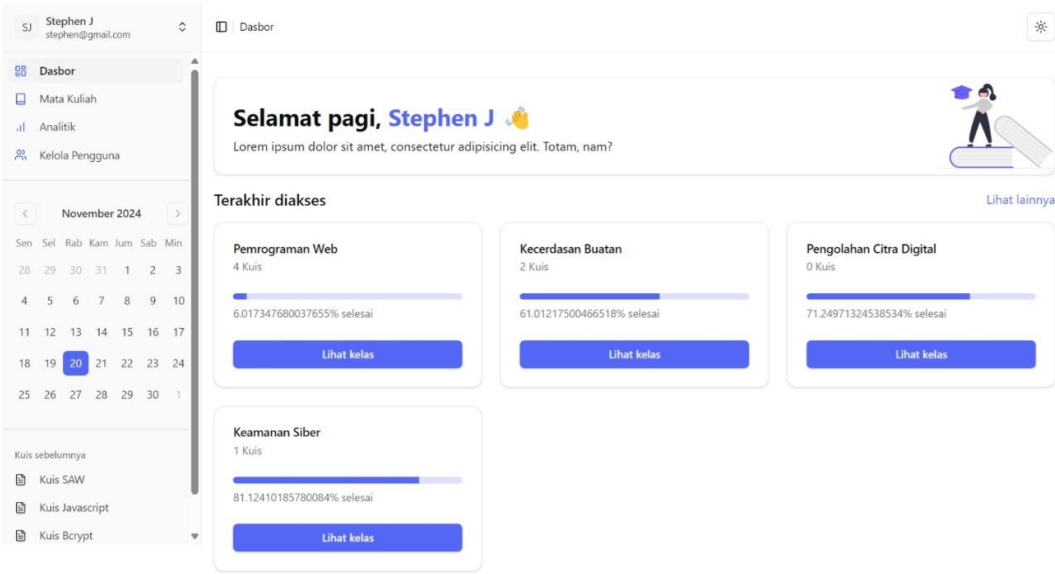
Daftar

Sudah punya akun? [Masuk](#)

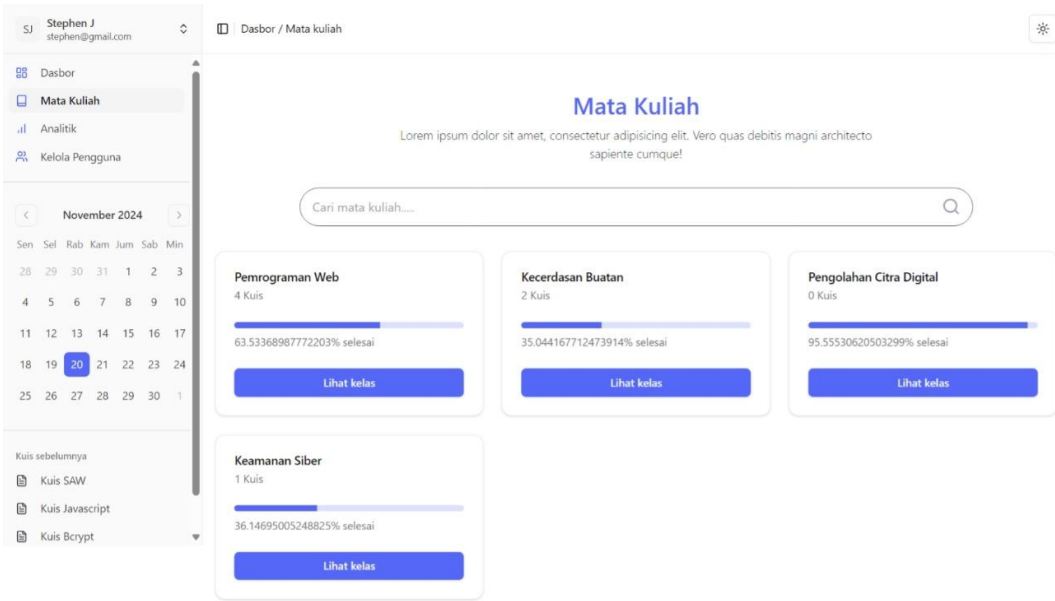
Kembali ke Beranda



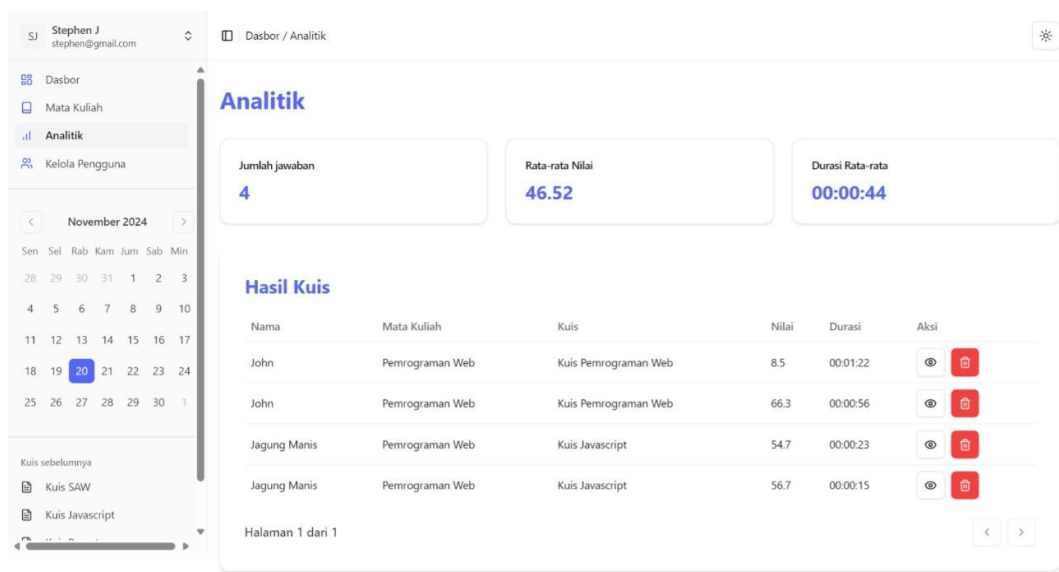
Gambar 4.3 Signup Page



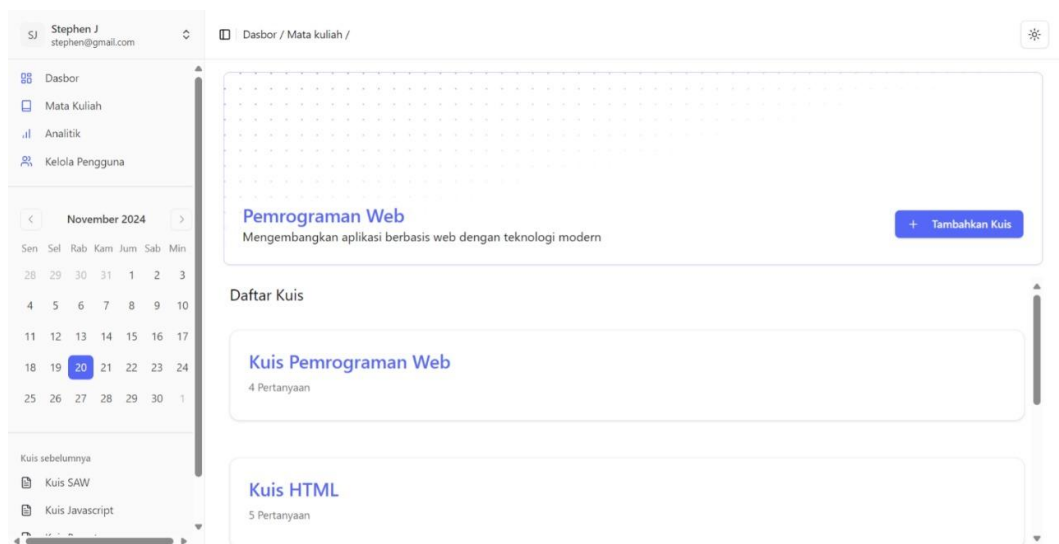
Gambar 4.4 Dashboard



Gambar 4.5 Tampilan Pilihan Mata Kuliah



Gambar 4.6 Tampilan Menu Analitik



Gambar 4.7 Tampilan Menu Menambah Kuis

Kuis Javascript

Soal 3 / 3 00:01:23

elaskan berbagai tipe data dalam JavaScript, seperti string, number, boolean, object, array, dan bagaimana mereka digunakan dalam pemrograman.

Dalam JavaScript, terdapat beberapa tipe data dasar yang sering digunakan dalam pemrograman. Berikut adalah penjelasan tentang beberapa tipe data utama beserta penggunaannya:


1. String
Tipe data string digunakan untuk menyimpan teks atau karakter. String dapat dibuat dengan tanda kutip tunggal (') atau ganda (").
2. Number

Jumlah kata: 47 Rekomendasi: 100-200 Kata

< Sebelumnya
Lihat semua
Serahkan >

1
2
3

Gambar 4.8 Tampilan Menu Menjawab Kuis


 Pemrograman Web
Kuis Javascript Selesai!
 Terima kasih sudah mengerjakan kuis, Jagung Manis!

Nilai anda :
74.3/100

Jumlah Pertanyaan
3

Durasi Pengerjaan
0j : 16m : 29d

Pertanyaan 1 v

Pertanyaan 2 v

Pertanyaan 3 ^

Pertanyaan:

elaskan berbagai tipe data dalam JavaScript, seperti string, number, boolean, object, array, dan bagaimana mereka digunakan dalam pemrograman.

Jawaban Anda:

JavaScript memiliki berbagai tipe data yang digunakan untuk menyimpan dan memanipulasi nilai. Primitive types mencakup string, number, boolean, null, undefined, bigint, dan symbol. Tipe string digunakan untuk menyimpan teks, didefinisikan menggunakan tanda kutip tunggal (') ganda ("), atau backtick (`) untuk template literal. Number digunakan untuk angka baik bilangan bulat maupun desimal. Boolean merepresentasikan nilai logika true atau false. Null adalah nilai khusus yang berarti "tidak ada nilai", sedangkan undefined adalah nilai default untuk variabel yang belum diinisialisasi. BigInt memungkinkan representasi angka yang sangat besar, melampaui batas number. Symbol menciptakan nilai unik yang biasanya digunakan sebagai identifier. Selain itu, JavaScript memiliki tipe data non-primitive seperti object dan array. Object adalah struktur data berbasis pasangan key-value, berguna untuk merepresentasikan entitas dengan berbagai atribut. Contohnya: {name: 'John', age: 30}. Array adalah struktur data yang digunakan untuk menyimpan sekumpulan nilai dalam urutan tertentu, misalnya [1, 2, 3]. Objek dan array sering digunakan bersama untuk menyimpan dan mengelola data yang kompleks. Penggunaan tipe data ini mempermudah manipulasi data, penghitungan, dan interaksi logis dalam program JavaScript, baik untuk aplikasi sederhana maupun kompleks.

Umpan Balik:

Kualitas esai sangat baik dengan sedikit aspek yang perlu disempurnakan.

% Nilai untuk jawaban ini:

77

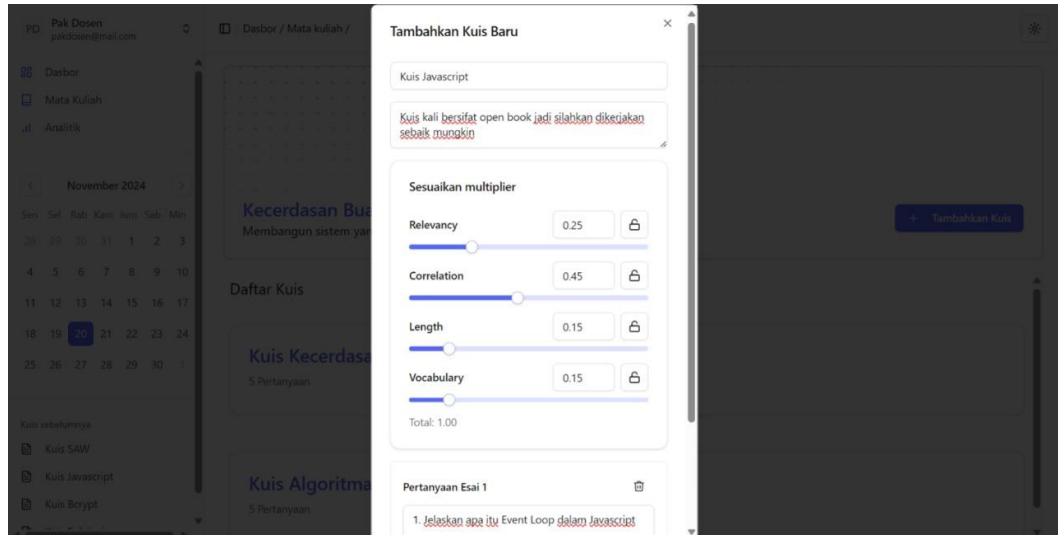
☆ Parameter penilaian:

Relevancy: 71
 Length: 81

Correlation: 82
 Vocabulary: 71

Kembali

Gambar 4.9 Tampilan Hasil Kuis



Gambar 4.10 Tampilan Menu Pengaturan Bobot Penilaian

4.3 Pre-processing Dataset

Pada tahap ini, dilakukan proses penghapusan karakter khusus menggunakan *library regular expression* (regex). Berikut adalah implementasi penghapusan karakter khusus pada *essay*.

```
[ ] def clean_essay(essay):
    if not isinstance(essay, str):
        essay = str(essay)
    essay = re.sub(r'^\x00-\x7F+', ' ', essay)
    essay = re.sub(r'\s+', ' ', essay).strip()
    return essay

df['essay'] = df['essay'].apply(clean_essay)
```

Gambar 4.11 Penghapusan Karakter Khusus

Fungsi `re.sub` yang digunakan pada penelitian ini adalah

1. `re.sub(r'^\x00-\x7F+', ' ', essay)` digunakan untuk mengganti semua karakter non-ASCII dengan spasi.
2. `re.sub(r'\s+', ' ', essay).strip()` digunakan untuk menghapus spasi yang berlebihan di dalam teks dan di awal/akhir kalimat.

4.4 Pengembangan Model

Pada tahap ini, kolom *prompt* dan *essay* diubah tipe datanya menjadi *string*. Setelah itu, *prompt* digabungkan dengan *essay* yang sudah dibersihkan menggunakan *separator* [SEP] dan di simpan dalam kolom baru bernama *combine*.

```
[ ] df['prompt'] = df['prompt'].astype('string')
    df['essay'] = df['essay'].astype('string')
```

Gambar 4.12 Pengubahan tipe data menjadi *string*

```
[ ] df['combine'] = df['prompt'] + ' [SEP] ' + df['essay']
```

Gambar 4.13 Penggabungan kolom *prompt* dan *essay*

Tahap selanjutnya yaitu melakukan skalasi untuk nilai *total_hscore* ke dalam rentang 0 dan 1 menggunakan *MinMaxScaler*. Untuk mendapatkan *relevancy_score* dan *correlation_score*, nilai yang sudah dinormalisasi kemudian dikali dengan bobotnya masing-masing.

```
[ ] scaler = MinMaxScaler(feature_range=(0, 1))
    df['normalized_score'] = scaler.fit_transform(df[['total_hscore']])

    df['relevancy_score'] = df['normalized_score'] * 0.8
    df['correlation_score'] = df['normalized_score'] * 0.9

    y_scores = df[['relevancy_score', 'correlation_score']].values
```

Gambar 4.14 Pengskalaan nilai holistik

Kemudian, dilakukan pemrosesan data menggunakan *tokenizer* IndoBERT, yaitu *indobenchmark/indobert-base-p2*. Setiap teks kemudian diproses menggunakan metode *tokenizer* dengan parameter seperti *truncation* untuk memotong teks yang melebihi panjang maksimum 512 token dan menambahkan *padding* untuk mengisi teks yang lebih pendek. *Output* dari *tokenizer* mencakup *input_ids*, yang merupakan representasi token numerik dan *attention_mask*, yang menunjukkan token mana yang harus diperhatikan atau diabaikan oleh model.

```
[ ] tokenizer = BertTokenizer.from_pretrained('indobenchmark/indobert-base-p2')
    encodings = tokenizer(df['combine'].tolist(), truncation=True, padding='max_length', max_length=512, return_tensors='tf')
```

Gambar 4.15 Tokenisasi teks

Setelah itu, diambil [CLS] yang merupakan token pertama dari setiap *input* dari lapisan terakhir model *last_hidden_state[:, 0, :]*. Representasi ini berfungsi sebagai *input* untuk lapisan *neural network* tambahan. Lapisan tambahan ini terdiri dari tiga buah lapisan *Dense*. Lapisan pertama terdapat 128 unit *neuron* yang ditambahkan fungsi aktivasi ReLU, fungsinya yaitu untuk menangkap pola non-linear dari representasi teks. Lapisan kedua memiliki 64 unit *neuron*, juga

menggunakan fungsi aktivasi ReLU untuk memproses informasi lebih lanjut. Akhirnya, lapisan terakhir terdiri dari dua unit *neuron* dengan fungsi aktivasi linear, yang digunakan untuk menghasilkan prediksi skor relevansi dan korelasi sebagai nilai kontinu. Model ini kemudian dirancang dengan menggunakan API Keras, dengan *input_ids* dan *attention_mask* sebagai *input*, dan dua skor prediksi sebagai *output*.

```
[ ] x = Dense(128, activation='relu')(cls_embedding)
    x = Dense(64, activation='relu')(x)
    output = Dense(2, activation='linear')(x)
```

Gambar 4.16 Lapisan tambahan *neural network*

Model ini dikompilasi menggunakan optimasi Adam dengan *learning rate* 2×10^{-5} yang telah diatur untuk stabilitas pelatihan pada model *pre-trained* seperti BERT. Fungsi kerugian (*loss*) yang digunakan adalah *Mean Squared Error* (MSE). Selain itu, metrik evaluasi *Mean Squared Error* juga digunakan untuk memantau performa model selama pelatihan.

```
[ ] model = Model(inputs=[input_ids, attention_mask], outputs=output)

optimizer = tf.keras.optimizers.Adam(learning_rate=2e-5)
loss = tf.keras.losses.MeanSquaredError()
model.compile(optimizer=optimizer, loss=loss, metrics=[tf.keras.metrics.MeanSquaredError()])
```

Gambar 4.17 Pembuatan model

Lalu, *input_ids* dan *attention_mask* dikonversi dari *tensor* menjadi *array*.

```
[ ] input_ids = encodings['input_ids'].numpy()
    attention_mask = encodings['attention_mask'].numpy()
```

Gambar 4.18 Konversi *tensor* menjadi *array*

Data yang telah ditokenisasi kemudian dipisahkan menjadi set pelatihan (80%) dan validasi (20%) menggunakan *train_test_split*. Pada langkah ini, baik *input_ids* maupun *attention_mask* diproses untuk masing-masing set, sehingga memastikan data pelatihan dan validasi memiliki struktur yang sesuai.

```
[ ] X_train_ids, X_val_ids, y_train, y_val = train_test_split(input_ids, y_scores, test_size=0.2, random_state=42)
    X_train_mask, X_val_mask, _, _ = train_test_split(attention_mask, y_scores, test_size=0.2, random_state=42)

X_train = {'input_ids': X_train_ids, 'attention_mask': X_train_mask}
X_val = {'input_ids': X_val_ids, 'attention_mask': X_val_mask}
```

Gambar 4.19 Pemisahan dataset *training* dan *validation*

Data tersebut kemudian diorganisasikan ke dalam dataset TensorFlow, di mana data pelatihan diacak (*shuffle*) dan dibagi menjadi *batch* dengan ukuran 16 untuk meningkatkan efisiensi pelatihan.

```
[ ] train_dataset = tf.data.Dataset.from_tensor_slices((dict(X_train), y_train)).shuffle(1000).batch(16)
    val_dataset = tf.data.Dataset.from_tensor_slices((dict(X_val), y_val)).batch(16)
```

Gambar 4.20 Pengaturan *batch size* pelatihan

Pelatihan model menggunakan metode *fit*, dengan *callback EarlyStopping* untuk mencegah *overfitting*. *Callback* ini memantau kerugian pada set validasi (*val_loss*) dan akan menghentikan pelatihan jika tidak ada perbaikan selama tiga *epoch* berturut-turut. Selain itu, bobot terbaik selama pelatihan akan dipulihkan (*restore_best_weights*). Model dilatih hingga *epoch* ke 10, namun ketika model mencapai *epoch* 7, sudah terlihat tanda-tanda terjadinya *overfitting*. Oleh karena itu, pelatihan dihentikan pada *epoch* ke 7.

```
[ ] early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True, verbose=1)
```

Gambar 4.21 Penggunaan metode *early stopping*

```
[ ] history = model.fit(train_dataset, epochs=10, validation_data=val_dataset, callbacks=[early_stopping])
```

Epoch 1/10
WARNING:tensorflow:Gradients do not exist for variables ['tf_bert_model/bert/pooler/dense/kernel:0', 'tf_bert_model/bert/pooler/dense/bias:0'] when minimizing the loss. If you
WARNING:tensorflow:Gradients do not exist for variables ['tf_bert_model/bert/pooler/dense/kernel:0', 'tf_bert_model/bert/pooler/dense/bias:0'] when minimizing the loss. If you
500/500 [=====] - 5944s 12s/step - loss: 0.0318 - mean_squared_error: 0.0318 - val_loss: 0.0090 - val_mean_squared_error: 0.0090
Epoch 2/10
500/500 [=====] - 5905s 12s/step - loss: 0.0103 - mean_squared_error: 0.0103 - val_loss: 0.0090 - val_mean_squared_error: 0.0090
Epoch 3/10
500/500 [=====] - 5937s 12s/step - loss: 0.0060 - mean_squared_error: 0.0060 - val_loss: 0.0068 - val_mean_squared_error: 0.0068
Epoch 4/10
500/500 [=====] - 5935s 12s/step - loss: 0.0038 - mean_squared_error: 0.0038 - val_loss: 0.0043 - val_mean_squared_error: 0.0043
Epoch 5/10
500/500 [=====] - 5909s 12s/step - loss: 0.0034 - mean_squared_error: 0.0034 - val_loss: 0.0040 - val_mean_squared_error: 0.0040
Epoch 6/10
500/500 [=====] - 5901s 12s/step - loss: 0.0029 - mean_squared_error: 0.0029 - val_loss: 0.0047 - val_mean_squared_error: 0.0047
Epoch 7/10
500/500 [=====] - 5905s 12s/step - loss: 0.0024 - mean_squared_error: 0.0024 - val_loss: 0.0096 - val_mean_squared_error: 0.0096
Epoch 8/10
107/500 [=====] - ETA: 1:11:37 - loss: 0.0020 - mean_squared_error: 0.0020
KeyboardInterrupt Traceback (most recent call last)
<ipython-input-16-9694e26d7409> in <cell line: 1>()

Gambar 4.22 Pelatihan model

4.5 Hyperparameter Fine-tuning

Hyperparameter yang digunakan untuk melatih model ini yaitu *learning rate* sebesar 2×10^{-5} , *batch size* 16, dengan 7 *epoch* dan menggunakan *optimizer* Adam.

4.6 Evaluasi

Dalam pelatihan ini, digunakan metrik evaluasi MSE (*Mean Squared Error*) pada data *validation* dan *Quadratic Weighted Kappa* (QWK). Dari hasil MSE yang didapat, yaitu sebesar 0.00335 menunjukkan bahwa model memiliki performa yang bagus dalam memprediksi nilai. Dari hasil QWK yang didapat,

0.9222 menunjukkan bahwa model memiliki tingkat kesesuaian yang sangat tinggi antara prediksi dan nilai sebenarnya pada skala ordinal.

```
[ ] mse_score = model.evaluate(val_dataset)
    print(f"MSE score on the validation set: {mse_score[1]}")

125/125 [=====] - 448s 4s/step - loss: 0.0034 - mean_squared_error: 0.0034
MSE score on the validation set: 0.003356082132086158
```

Gambar 4.23 Hasil evaluasi dengan MSE

```
from sklearn.metrics import cohen_kappa_score

qwk = cohen_kappa_score(df['true_score'], df['predicted_score'], weights="quadratic")
print(f"Quadratic Weighted Kappa (QWK): {qwk}")

Quadratic Weighted Kappa (QWK): 0.9222449972823514
```

Gambar 4.24 Hasil evaluasi dengan QWK

4.7 Rubrik Penilaian

Terdapat empat rubrik penilaian yang digunakan dalam penilaian esai yaitu relevansi jawaban dengan *prompt*, korelasi antar kalimat, panjang esai, dan kekayaan kosakata. Rubrik penilaian semantik seperti relevansi dan korelasi akan di prediksi oleh model, sedangkan rubrik penilaian sintaksis seperti panjang esai dan kekayaan kosakata ditambahkan secara eksplisit, menggunakan rumus yang sudah dijelaskan sebelumnya.

1. Relevansi jawaban dengan *prompt*

Nilai relevansi diprediksi oleh model berdasarkan proses pelatihan yang sudah dijelaskan sebelumnya.

2. Korelasi antar kalimat

Sama dengan nilai relevansi, nilai ini juga diprediksi oleh model berdasarkan proses pelatihan.

3. Panjang esai

Berdasarkan rumus diatas, panjang esai dikalkulasikan dengan melihat jumlah katanya, semakin panjang jumlah katanya, semakin tinggi pula nilai yang didapatkan.

```
[ ] def calculate_length_score(essay):
    word_count = len(essay.split())
    if word_count >= 200:
        return 100
    elif word_count >= 100:
        return round(50 + 40 * (1 - np.exp(-0.02 * (word_count - 100))))
    else:
        return round(25 + 25 * (1 - np.exp(-0.04 * word_count)))
```

Gambar 4.25 Penilaian panjang esai

Pertama, fungsi ini menghitung jumlah kata dengan menggunakan `essay.split()` yang membagi teks berdasarkan spasi. Nilai dihitung berdasarkan ketentuan pada tabel berikut.

Tabel 4.1 Perhitungan penilaian panjang esai

Jumlah kata	Nilai
≥ 200	100
100 – 200	$50 + 40 \times (1 - e^{-0.02 \times (\text{word_count} - 100)})$
< 100	$25 + 25 \times (1 - e^{-0.04 \times \text{word_count}})$

4. Kekayaan kosakata

Nilai kekayaan kosakata dihitung menggunakan *Type-Token Ratio* (TTR). Semakin banyak kata yang berbeda dalam teks, semakin tinggi TTR dan semakin tinggi pula skor kosakata.

```
[ ] def calculate_vocabulary_score(essay):
    words = essay.split()
    unique_words = set(words)
    ttr = len(unique_words) / len(words) if len(words) > 0 else 0
    adjusted_ttr = ttr * (len(words) / 50) if len(words) < 50 else ttr
    return round(adjusted_ttr * 100)
```

Gambar 4.26 Penilaian kekayaan kosakata

Nilai perhitungan kekayaan kosakata yaitu:

Tabel 4.2 Perhitungan penilaian kekayaan kosakata

Panjang teks	Nilai TTR
≥ 50	Jumlah kata unik/jumlah kata
< 50	$(\text{Jumlah kata unik/jumlah kata}) \times (\text{Jumlah kata}/50)$

Penyesuaian rumus ini dilakukan untuk memberikan bobot lebih pada esai yang lebih pendek, karena pada teks yang lebih pendek, tingkat variasi kata dapat lebih rendah. Hasilnya, skor kosakata dikali 100, untuk mengubah nilai TTR menjadi skor dalam rentang 0-100.

4.8 Feedback

Pada tahap ini, model akan menghasilkan umpan balik yang komprehensif berdasarkan hasil evaluasi nilai esai dari keempat aspek. Setelah mendapatkan nilai prediksi *relevancy_score* dan *correlation_score* dari model, fungsi ini

kemudian menghitung nilai panjang esai dan kekayaan kosakata menggunakan dua fungsi yang telah dijelaskan sebelumnya. Semua skor ini kemudian digabungkan untuk menghasilkan *total_ascore* dengan rumus pembobotan secara *default* yaitu:

$$\begin{aligned} total_ascore = & (relevancy_score \times 0.25) + \\ & (correlation_score \times 0.45) + (length_score \times 0.15) + \\ & (vocabulary_score \times 0.15) \end{aligned} \quad (6)$$

Pembobotan ini dirancang untuk menekankan korelasi antar kalimat sebagai faktor yang lebih penting, diikuti oleh relevansi jawaban dengan *prompt*, dan dengan sedikit bobot pada panjang esai dan kekayaan kosakata. Namun, pembobotan ini bisa disesuaikan kembali oleh masing-masing pengajar.

Kemudian, berdasarkan hasil tiap parameter penilaian dikategorikan kedalam *Good*, *Average* atau *Poor*. *Feedback* kemudian akan diberikan sesuai dengan jumlah kelas yang didapatkan.

```
feedbacks = {
    "relevancy_feedback": "Good" if relevancy_score > 70 else "Average" if relevancy_score >= 40 else "Poor",
    "correlation_feedback": "Good" if correlation_score > 80 else "Average" if correlation_score >= 50 else "Poor",
    "length_feedback": "Good" if length_score > 90 else "Average" if length_score >= 66 else "Poor",
    "vocabulary_feedback": "Good" if vocabulary_score > 70 else "Average" if vocabulary_score >= 50 else "Poor"
}

good_count = list(feedbacks.values()).count("Good")
average_count = list(feedbacks.values()).count("Average")
poor_count = list(feedbacks.values()).count("Poor")
```

Gambar 4.27 Pembagian kategori setiap parameter

```
if good_count == 4:
    feedback = "Kualitas esai luar biasa, memenuhi ketentuan di semua aspek."
elif good_count == 3 and average_count == 1:
    feedback = "Kualitas esai sangat baik dengan sedikit aspek yang perlu disempurnakan."
elif good_count == 3 and poor_count == 1:
    feedback = "Kualitas esai yang bagus, meskipun ada beberapa aspek yang bisa ditingkatkan lagi."
elif good_count == 2 and average_count == 2:
    feedback = "Secara keseluruhan kualitas esai baik, namun masih terdapat ruang untuk pengembangan."
elif good_count == 2 and average_count == 1 and poor_count == 1:
    feedback = "Kualitas esai cukup baik, meskipun beberapa aspek mengurangi kualitasnya secara keseluruhan."
elif good_count == 2 and poor_count == 2:
    feedback = "Kualitas esai masih dapat diterima, namun memerlukan perbaikan khusus."
elif good_count == 1 and average_count == 3:
    feedback = "Kualitas esai sedang, akan lebih baik dengan peningkatan di berbagai bidang."
elif good_count == 1 and average_count == 2 and poor_count == 1:
    feedback = "Kualitas esai terbatas karena beberapa aspek pada esai sangat terbatas."
elif good_count == 1 and average_count == 1 and poor_count == 2:
    feedback = "Kualitas esai kurang memadai, beberapa bagian membutuhkan perbaikan signifikan."
elif good_count == 1 and poor_count == 3:
    feedback = "Kualitas esai tidak memadai, masih terdapat banyak kekurangan."
elif average_count == 4:
    feedback = "Kualitas esai pada tingkat biasa, konsisten pada tingkat rata-rata."
elif average_count == 3 and poor_count == 1:
    feedback = "Kualitas esai cukup memadai, namun ada kelemahan yang bisa diperbaiki."
elif average_count == 2 and poor_count == 2:
    feedback = "Kualitas esai tidak memuaskan, masih kurang pada aspek-aspek penting."
elif average_count == 1 and poor_count == 3:
    feedback = "Kualitas esai sangat buruk, perlu banyak perbaikan signifikan untuk menghasilkan esai yang bagus."
elif poor_count == 4:
    feedback = "Esai tidak dapat diterima, kualitasnya jauh dari kata sempurna karena gagal memenuhi standar dasar pada semua aspek."
else:
    feedback = "Esai membutuhkan peningkatan yang signifikan."
```

Gambar 4.28 Pemberian *feedback*

4.9 Pengujian

Proses pengujian pada sistem ini dilakukan dengan menguji beberapa pasangan pertanyaan dan jawaban. *Output* dari hasil pengujian yaitu berupa nilai setiap parameter penilaian, *total_ascore*, serta *feedback*.

Tabel 4.3 Pengujian pertama

Prompt	Jelaskan apa yang dimaksud dengan perangkat lunak!
Essay	<p>Perangkat lunak merupakan kumpulan perintah yang digunakan oleh komputer untuk menjalankan berbagai fungsi atau tugas tertentu. Secara umum, ia tidak nampak secara fisik dan tidak dapat dipegang seperti perangkat keras, namun sangat krusial karena dapat mengendalikan dan mengoperasikan perangkat keras.</p> <p>Secara teknis, perangkat lunak terdiri dari dua kategori utama: aplikasi dan sistem. Perangkat lunak aplikasi dirancang untuk memudahkan tugas manusia, seperti mengolah kata bisa menggunakan aplikasi Microsoft Word, mengolah data menggunakan Excel, desain grafis menggunakan Photoshop.</p> <p>Sementara itu, perangkat lunak sistem berfungsi untuk mengelola dan mengatur perangkat keras komputer serta menyediakan platform bagi perangkat lunak aplikasi untuk dijalankan. Contohnya adalah Windows, macOS dan Linux. Sistem operasi ini mengatur semua sumber daya perangkat keras komputer, seperti pengolahan data, memori dan input/output.</p> <p>Selain perangkat lunak aplikasi dan sistem, ada juga perangkat lunak pengembangan (development software), yang meliputi alat-alat yang digunakan oleh programmer untuk membuat perangkat lunak lain, seperti compiler, editor teks, dan lingkungan pengembangan terintegrasi (IDE).</p> <p>Pengembangan perangkat lunak melibatkan perencanaan, desain, pengkodean, pengujian, dan pemeliharaan. Setiap perangkat lunak harus menjalani fase pengujian untuk dipastikan bebas dari kesalahan atau bug. Selain itu, perangkat lunak juga perlu diperbarui secara berkala untuk meningkatkan kinerja, menambah fitur baru, atau memperbaiki kelemahan yang ditemukan.</p>

	Secara keseluruhan, perangkat lunak adalah elemen yang sangat vital dalam ekosistem teknologi informasi karena tanpanya, perangkat keras (hardware) komputer tidak dapat berfungsi sesuai keinginan pengguna.
<i>Relevancy_score</i>	74
<i>Correlation_score</i>	86
<i>Length_score</i>	100
<i>Vocabulary_score</i>	64
<i>Total_ascore</i>	81
<i>Feedback</i>	Kualitas esai sangat baik dengan sedikit aspek yang perlu disempurnakan.

Pada pengujian pertama, diuji dengan jawaban yang panjang, dan didapatkan nilai relevansi 74, nilai korelasi 86, nilai panjang esai 100 dan nilai kosakata 64. Keempat nilai tersebut kemudian dikalikan dengan bobotnya masing-masing. Secara *default*, bobot nilai relevansi 0.25, bobot nilai korelasi 0.45, bobot nilai panjang esai 0.15 dan bobot kosakata 0.15. Dengan demikian, didapatkan nilai *total_ascore* 81 dan *feedback* kualitas esai sangat baik dengan sedikit aspek yang perlu disempurnakan.

Tabel 4.4 Pengujian kedua

<i>Prompt</i>	Apa itu kura-kura?
<i>Essay</i>	Kura-kura adalah hewan berkaki empat yang termasuk dalam golongan reptil. Kura-kura memiliki ciri khas tubuh yang dilindungi oleh cangkang keras yang berfungsi sebagai perlindungan dari predator. Hewan ini dapat ditemukan di banyak tempat, baik di daratan maupun di perairan. Kura-kura memiliki berbagai jenis, seperti kura-kura darat dan kura-kura air. Kura-kura juga dikenal sebagai hewan yang sangat lambat bergerak, namun memiliki ketahanan hidup yang luar biasa.
<i>Relevancy_score</i>	51
<i>Correlation_score</i>	61
<i>Length_score</i>	48
<i>Vocabulary_score</i>	78
<i>Total_ascore</i>	59
<i>Feedback</i>	Kualitas esai terbatas karena beberapa aspek pada esai sangat terbatas.

Pada pengujian kedua, digunakan esai yang lebih pendek, namun jawabannya masih ada relevansinya dengan pertanyaan. Hasilnya didapatkan nilai relevansi 51, nilai korelasi 61, nilai panjang esai 48 dan nilai kosakata 78. Bobot yang digunakan untuk pengujian kedua ini menggunakan bobot default, dan didapatkan nilai *total_ascore* 59, serta feedbacknya yaitu kualitas esai terbatas karena beberapa aspek pada esai sangat terbatas.

Tabel 4.5 Pengujian ketiga

<i>Prompt</i>	Jelaskan tentang revolusi industri.
<i>Essay</i>	Rendang adalah makanan yang dimakan oleh manusia purba. Di zaman ini, rendang adalah makanan pokok. Seiring berkembangnya zaman dan terjadinya revolusi industri, rendang sudah mulai ditinggalkan oleh manusia.
<i>Relevancy_score</i>	5
<i>Correlation_score</i>	0
<i>Length_score</i>	42
<i>Vocabulary_score</i>	46
<i>Total_ascore</i>	14
<i>Feedback</i>	Esai tidak dapat diterima, kualitasnya jauh dari kata sempurna karena gagal memenuhi standar dasar pada semua aspek.

Pada pengujian ketiga, digunakan jawaban yang pendek namun jawaban tersebut tidak berhubungan dengan pertanyaan yang diberikan. Didapatkan nilai relevansi 5, nilai korelasi 0, nilai panjang esai 42 dan nilai kosakata 46. Nilai *total_ascore* yang didapatkan melalui perkalian dengan bobot *default* mendapatkan nilai 14 dan *feedback* esai tidak dapat diterima, kualitasnya jauh dari kata sempurna karena gagal memenuhi standar dasar pada semua aspek.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil penerapan model IndoBERT untuk menilai esai secara otomatis adalah:

1. Model yang dikembangkan menggunakan teknik *fine-tuning* IndoBERT terbukti efektif dalam menangani tugas penilaian esai otomatis. Dengan memanfaatkan arsitektur *Transformer* dan mekanisme *self-attention*, model mampu memahami hubungan semantik antar teks.
2. Tahapan *pre-processing* dataset, termasuk normalisasi data, tokenisasi, dan pembersihan teks esai, sangat berpengaruh dalam meningkatkan performa model. Proses ini membantu mengurangi nilai loss selama pelatihan model, sehingga menghasilkan prediksi yang lebih akurat dan stabil.
3. Penggunaan rubrik penilaian berbasis sintaksis dan semantik, seperti relevansi jawaban dengan *prompt*, korelasi antar kalimat, panjang esai, dan kekayaan kosakata, memberikan kerangka evaluasi yang terukur dan objektif.
4. Metrik MSE digunakan untuk mengevaluasi model, karena metrik ini cocok digunakan untuk tugas regresi. Selain itu, juga digunakan metrik QWK untuk melihat kesesuaian nilainya. Hasil evaluasi menunjukkan bahwa model dapat memprediksi skor dengan tingkat kesalahan yang rendah.
5. Implementasi model ini menunjukkan potensi besar dalam mendukung tugas penilaian esai. Sistem ini mampu memberikan hasil evaluasi yang konsisten dan membantu mengurangi subjektivitas yang sering terjadi dalam penilaian manual.

5.2 Saran

Penulis memberikan beberapa saran yang dapat dilakukan pada penelitian kedepannya yaitu:

1. Pengembangan sistem penilaian esai otomatis yang mampu menangani tugas berbasis matematis.
2. Perbanyak dataset penelitian agar meningkatkan performa model.

DAFTAR PUSTAKA

- Beseiso, M., Alzubi, O. A., & Rashaideh, H. (2021). A novel automated essay scoring approach for reliable higher educational assessments. *Journal of Computing in Higher Education*, 33(3), 727–746. <https://doi.org/10.1007/s12528-021-09283-1>
- Buditjahjanto, I. G. P. A., Idhom, M., Munoto, M., & Samani, M. (2022). An Automated Essay Scoring Based on Neural Networks to Predict and Classify Competence of Examinees in Community Academy. *TEM Journal*, 11(4), 1694–1701. <https://doi.org/10.18421/TEM114-34>
- Chapman, A., Simperl, E., Koesten, L., Konstantinidis, G., Ibáñez, L. D., Kacprzak, E., & Groth, P. (2020). Dataset search: a survey. *VLDB Journal*, 29(1), 251–272. <https://doi.org/10.1007/s00778-019-00564-x>
- Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Naacl-Hlt 2019, Mlm*, 4171–4186. <https://aclanthology.org/N19-1423.pdf>
- Djoko, E. R., Rikel, ;, Mansor, M. ;, & Slater, R. (2020). Other Computer Engineering Commons, and the Technology and Innovation Commons Recommended Citation Recommended Citation Fu. *SMU Data Science Review*, 3(3),3.<https://scholar.smu.edu/datasciencereview>Availableat:<https://scholar.smu.edu/datasciencereview/vol3/iss3/3http://digitalrepository.smu.edu>
- Faradhila, A. U. T. (2024). Automated Essay Scoring dengan Fitur Feedback pada Essay Bahasa Indonesia Berbasis BERT.
- Hambali, B., Ma'mun, A., Susetyo, B., Hidayat, Y., & Gumilar, A. (2022). Validitas dan Reliabilitas Rubrik Penilaian Tes Hasil Belajar Keterampilan High Service Untuk Siswa Sekolah Dasar. *TEGAR: Journal of Teaching Physical Education in Elementary School*, 5(2). <https://doi.org/10.17509/tegar.v5i2.46208>
- Hussein, M. A., Hassan, H. A., & Nassef, M. (2020). A trait-based deep learning automated essay scoring system with adaptive feedback. *International Journal of Advanced Computer Science and Applications*, 11(5), 287–293. <https://doi.org/10.14569/IJACSA.2020.0110538>
- Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. *ACM Computing Surveys*, 54(10), 1–30. <https://doi.org/10.1145/3505244>

- Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3), 3713–3744. <https://doi.org/10.1007/s11042-022-13428-4>
- Koroteev, M. V. (2021). *BERT: A Review of Applications in Natural Language Processing and Understanding*. <http://arxiv.org/abs/2103.11943>
- Kumar, V. S., & Boulanger, D. (2021). Automated Essay Scoring and the Deep Learning Black Box: How Are Rubric Scores Determined? *International Journal of Artificial Intelligence in Education*, 31(3), 538–584. <https://doi.org/10.1007/s40593-020-00211-5>
- Mizumoto, A., & Eguchi, M. (2023). Exploring the potential of using an AI language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2), 100050. <https://doi.org/10.1016/j.rmal.2023.100050>
- Nadeem, F. (2022). Evaluating and Ranking Cloud IaaS, PaaS and SaaS Models Based on Functional and Non-Functional Key Performance Indicators. *IEEE Access*, 10, 63245–63257. <https://doi.org/10.1109/ACCESS.2022.3182688>
- Noviani, Zubaidah, & Bistari. (2019). *Pengaruh Umpan Balik Pekerjaan Rumah Terhadap Rasa Tanggung Jawab Dan Hasil Belajar Matematika*. 1–8.
- Paton, N. W., Chen, J., Wu, Z., & Science, C. (2023). *Dataset Discovery and Exploration : A Survey*. 56(4). <https://doi.org/10.1145/3626521>
- Qasrawi, R., & Beniabdelrahman, A. (2020). The Higher And Lower-Order Thinking Skills (HOTS and LOTS) In Unlock English Textbooks (1st And 2nd Editions) Based On Bloom’S Taxonomy: An Analysis Study. *International Online Journal of Education and Teaching (IOJET)*, 7(3), 744–758. <https://iojet.org/index.php/IOJET/article/view/866>
- Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. In *Artificial Intelligence Review* (Vol. 55, Issue 3). Springer Netherlands. <https://doi.org/10.1007/s10462-021-10068-2>
- Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8, 842–866. https://doi.org/10.1162/tac1_a_00349
- Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2(6), 1–20. <https://doi.org/10.1007/s42979-021-00815-1>
- Singh, A., Pandey, N., Shirgaonkar, A., Manoj, P., & Aski, V. (2024). A Study of

- Optimizations for Fine-tuning Large Language Models. arXiv preprint arXiv:2406.02290.
- Terence, L. (2022). The Validity of Multiple Choice and Essay Composition Proficiency Assessment for English Language Learners. *Global Journal of English Language Teaching*, 2(2), 16–19. <https://doi.org/10.20448/gjelt.v2i2.4226>
- Uto, M., Xie, Y., & Ueno, M. (2020). Neural Automated Essay Scoring Incorporating Handcrafted Features. *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 6077–6088. <https://doi.org/10.5715/jnlp.28.716>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-Decem(Nips)*, 5999–6009.
- Wang, Y., & Li, R. (2022). *On the Use of BERT for Automated Essay Scoring : Joint Learning of Multi-Scale Essay Representation*. 3416–3425.
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>
- Yathiraju, N. (2022). Investigating the use of an Artificial Intelligence Model in an ERP Cloud-Based System. *International Journal of Electrical, Electronics and Computers*, 7(2), 01–26. <https://doi.org/10.22161/eec.72.1>
- Zhong, W. (2024). Effectiveness of finetuning pretrained BERT and deBERTa for automatic essay scoring. *Applied and Computational Engineering*, 52(1), 87–95. <https://doi.org/10.54254/2755-2721/52/20241321>