

***AUTOMATED ESSAY SCORING DENGAN FITUR FEEDBACK***  
**PADA ESSAY BAHASA INDONESIA BERBASIS BERT**

**SKRIPSI**

**FARADHILA AULIA UTAMI TANJUNG**

**201401005**



**PROGRAM STUDI S-1 ILMU KOMPUTER**  
**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**  
**UNIVERSITAS SUMATERA UTARA**  
**MEDAN**  
**2024**

***AUTOMATED ESSAY SCORING DENGAN FITUR FEEDBACK PADA  
ESSAY BAHASA INDONESIA BERBASIS BERT***

**SKRIPSI**

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

**FARADHILA AULIA UTAMI TANJUNG**

**201401005**



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

**MEDAN**

**2024**

**PERSETUJUAN**

Judul : *AUTOMATED ESSAY SCORING* (AES) DENGAN  
FITUR *FEEDBACK* PADA ESSAY BAHASA  
INDONESIA BERBASIS BERT

Kategori : SKRIPSI

Nama : FARADHILA AULIA UTAMI TANJUNG

Nomor Induk Mahasiswa : 201401005

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Pembimbing 2



Dewi Sartika Br. Ginting, S.Kom., M.Kom

NIP. 199005042019032023

Pembimbing 1



Dr. Amalia ST., M.T.

NIP. 197812212014042001

Diketahui/Disetujui Oleh

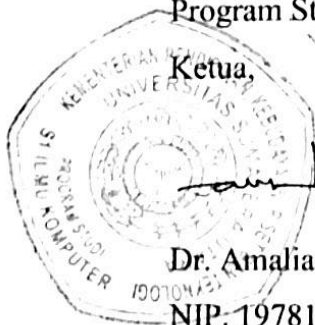
Program Studi S-1 Ilmu Komputer

Ketua,



Dr. Amalia ST., M.T.

NIP. 197812212014042001



**PERNYATAAN*****AUTOMATED ESSAY SCORING DENGAN FITUR FEEDBACK PADA ESSAY  
BAHASA INDONESIA BERBASIS BERT*****SKRIPSI**

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

Medan, 11 Juni 2024



Faradhila Aulia Utami Tanjung  
201401005

## PENGHARGAAN

*Bismillahirrahmanirrahim*, segala puji syukur dipanjatkan kepada Allah *Subhanahu Wa Ta'ala* atas segala limpahan rahmat dan hidayah-Nya sehingga penulis dapat berada di tahap penyusunan skripsi ini sebagai syarat untuk mendapatkan gelar Sarjana Komputer di Program Studi S-1 Ilmu Komputer, Universitas Sumatera Utara. Tidak lupa shalawat serta salam tetap tercurahkan kepada Rasulullah *Shalallaahu 'Alayhi Wasallam* yang telah mengeluarkan umat manusia dari kegelapan menuju zaman terang benderang saat ini.

Dengan penuh rasa hormat pada kesempatan ini penulis mengucapkan terima kasih kepada Ibu tercinta, Risdaliana Tanjung atas segala bentuk perjuangan, kasih sayang, dan perlindungan dengan doa-doa yang dipanjatkan untuk penulis. Dan terima kasih kepada Ayah, Chairul Anwar Tanjung atas dukungan dan kasih sayang yang membersamai di setiap langkah penulis. Terima kasih untuk setiap dukungan yang telah diberikan hingga penulis dapat berada di titik ini.

Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, dan bimbingan dari banyak pihak. Oleh karena itu, penulis mengucapkan banyak 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 penyusunan skripsi ini.
5. Ibu Dewi Sartika Br. Ginting S.Kom., M.Kom. selaku dosen Pembimbing II yang telah memberikan banyak bimbingan dan masukan yang berharga kepada penulis selama proses penyusunan skripsi ini.

6. Ibu Hayatunnufus S.Kom., M.Kom selaku Dosen Pembimbing Akademik yang telah memberikan saran, motivasi dan banyak dukungan kepada penulis selama perkuliahan.
7. Seluruh Bapak dan Ibu Dosen Program Studi S-1 Ilmu Komputer, yang telah membimbing saya selama masa perkuliahan hingga akhir masa studi.
8. Teristimewa kepada Orang Tua terkasih, tersayang penulis Chairul Anwar Tanjung dan Risdaliana Tanjung yang telah memberikan penulis kasih sayang yang tiada henti, ilmu yang bermanfaat, dan doa yang tiada henti bagi penulis sehingga penulis dapat menjalani perkuliahan dengan baik sampai dengan penyusunan skripsi.
9. Adik kandung tercinta Nayla Syifa Tanjung dan Ulfa Hasanah yang selalu mendukung dan mendoakan penuli dalam menjalani kehidupan masa kuliah hingga sampai menyelesaikan tugas akhir.
10. Keluarga Rumah Tinggi dan Merpati yang selalu percaya, menyemangati dan mendoakan penulis dalam setiap langkah yang diambil untuk mendapatkan gelar S-1.
11. Sahabat Papa And The Kids yaitu Syavira, Bebi, Rizka, Angel, Ronal, Erick dan Yudha yang telah menemani penulis dari kecil hingga penulis bisa menyelesaikan perkuliahan.
12. Sahabat SQD yang tidak bisa disebutkan satu-satu, telah memberikan penulis motivasi hingga nasehat dalam menyelesaikan proses pembuatan skripsi.
13. Sahabat Kureng Literasi yaitu Erlin, Devi, Difanie dan Tari yang telah kebersamai perjalanan kuliah, kehidupan perantauan dan proses pembuatan skripsi penulis.
14. Teman seperjuangan Kaum Mujahirudin yang telah memberikan banyak motivasi kepada penulis untuk menyelesaikan skripsi tepat waktu.
15. Atha Maulana dan keluarga yang telah menemani, memberikan semangat, motivasi dan bantuan kepada penulis selama masa perkuliahan, perantauan dan proses penyusunan skripsi.
16. Stambuk 2020 terkhusus kom A yang telah menemani penulis kuliah dan memberikan pengalaman belajar yang berharga kepada penulis.
17. Abang-kakak senior terkhusus stambuk 2018 yang telah memberikan banyak masukan, arahan, motivasi dan doa baik kepada penulis selama masa perkuliahan hingga penulisan skripsi ini.

18. Taylor Swift, NCT, Seventeen dan Treasure yang sangat memberikan motivasi kepada penulis untuk menyelesaikan perkuliahan dengan lagu dan konten yang dibuat.

Dan seluruh pihak yang telah memberi dukungan serta doa baik yang tidak dapat penulis sebutkan satu per-satu. Semoga Allah *Subhanahu Wa Ta'ala* senantiasa melimpahkan keberkahan serta kebaikan atas semua dukungan yang telah diberikan kepada penulis dan hasil penelitian ini dapat memberi manfaat maupun inspirasi untuk kedepannya.

Medan, 11 Juni 2024

Penulis,

A handwritten signature in black ink, consisting of stylized, overlapping loops and a vertical line extending downwards.

Faradhila Aulia Utami Tanjung

## ABSTRAK

Mengembangkan sistem Automated Essay Scoring (AES) yang efektif dan akurat untuk penilaian esai dalam bahasa Indonesia. Dalam konteks perkuliahan, Capaian Pembelajaran Mata Kuliah (CPMK) menjadi acuan bagi dosen untuk mengevaluasi kemampuan mahasiswa. Salah satu metode evaluasi yang umum digunakan adalah penilaian esai, namun proses ini memakan waktu lama dan memerlukan pengelompokan skor yang tepat untuk memberikan umpan balik yang sesuai. Dalam mengatasi tantangan ini, dibutuhkan sistem penilaian esai dan feedback secara otomatis dengan menggunakan pendekatan Bidirectional Encoder Representations for Transformers (BERT), yang dapat melakukan tugas klasifikasi sekuens untuk mendapatkan nilai sebenarnya dari sebuah esai. Penelitian ini memanfaatkan Word2Vec dalam melakukan representasi vektor untuk penilaian similarity antara jawaban esai dengan referensi esai yang telah ditetapkan. Dataset yang digunakan pada penelitian ini diambil dari kompetisi Kaggle yang berjudul The Hewlett Foundation: Automated Essay Scoring lalu diterjemahkan ke dalam bahasa Indonesia, dataset yang digunakan pada penelitian ini sebanyak 3.258 dataset yang berisi esai dan juga nilai yang diberikan manusia. Model base yang digunakan pada penelitian ini adalah model “bert-base-uncased” yang sudah dilatih dalam tugas klasifikasi teks. Model tersebut kemudian dilatih menggunakan dataset hingga mendapatkan hasil accuracy sempurna sebesar 1 dan nilai kappa score yang baik sebesar 0.82 yang kemudian model tersebut digunakan untuk melakukan prediksi nilai esai dengan text similarity, rubrik skor dan BERT. Hasil dari prediksi nilai tersebut memberikan nilai dan umpan balik yang sesuai dengan nilai akhir yang didapatkan penulis.

**Kata Kunci:** *Automated Essay Scoring, Natural Language Processing, BERT, Text Similarity, Word2Vec, Feedback*



## ABSTRACT

### AUTOMATED ESSAY SCORING WITH FEEDBACK FEATURE ON INDONESIA LANGUAGE ESSAY BASED ON BERT

Developing an effective and accurate Automated Essay Scoring (AES) system for assessing essays in Bahasa Indonesia is the focus of this research. In the context of higher education, Course Learning Outcomes (CLO) serve as a reference for instructors to evaluate students' abilities. One common evaluation method used is essay assessment, which, however, is time-consuming and requires proper score categorization to provide appropriate feedback. To address this challenge, an automated essay scoring and feedback system is needed, utilizing the Bidirectional Encoder Representations for Transformers (BERT) approach, capable of sequence classification tasks to obtain the true value of an essay. This research leverages Word2Vec for vector representation to assess similarity between essay answers and established reference essays. The dataset used in this study is sourced from the Kaggle competition titled "The Hewlett Foundation: Automated Essay Scoring," translated into Bahasa Indonesia, consisting of 3,258 essays and human-assigned scores. The base model utilized in this research is the "bert-base-uncased" model pre-trained on text classification tasks. This model is then fine-tuned using the dataset until achieving a perfect accuracy score of 1 and a satisfactory kappa score of 0.82. Subsequently, the model is employed to predict essay scores using text similarity, rubric scores, and BERT. The predicted scores provide appropriate feedback aligned with the author's final score.

**Keywords:** *Automated Essay Scoring, Natural Language Processing, BERT, Text Similarity, Word2Vec, Feedback*

## DAFTAR ISI

|   |             |
|---|-------------|
| <b>PERSETUJUAN .....</b>                      | <b>ii</b>   |
| <b>PERNYATAAN.....</b>                        | <b>iii</b>  |
| <b>PENGHARGAAN.....</b>                       | <b>iv</b>   |
| <b>ABSTRAK .....</b>                          | <b>vii</b>  |
| <b>ABSTRACT .....</b>                         | <b>viii</b> |
| <b>DAFTAR ISI.....</b>                        | <b>ix</b>   |
| <b>DAFTAR TABEL .....</b>                     | <b>xii</b>  |
| <b>DAFTAR GAMBAR.....</b>                     | <b>xiii</b> |
| <b>BAB 1 PENDAHULUAN .....</b>                | <b>1</b>    |
| <b>1.1 Latar Belakang.....</b>                | <b>1</b>    |
| <b>1.2 Rumusan Masalah.....</b>               | <b>6</b>    |
| <b>1.3 Batasan Masalah .....</b>              | <b>6</b>    |
| <b>1.4 Tujuan Penelitian.....</b>             | <b>7</b>    |
| <b>1.5 Manfaat Penelitian.....</b>            | <b>7</b>    |
| <b>1.6 Metodologi Penelitian .....</b>        | <b>7</b>    |
| <b>1.7 Penelitian Relevan.....</b>            | <b>8</b>    |
| <b>1.8 Sistematika Penulisan .....</b>        | <b>10</b>   |
| <b>BAB 2 LANDASAN TEORI .....</b>             | <b>12</b>   |
| <b>2.1 Automated Essay Scoring (AES).....</b> | <b>12</b>   |
| 2.1.1 Evaluasi Esai.....                      | 12          |
| <b>2.2 Rubrik Penilaian .....</b>             | <b>13</b>   |
| 2.2.1 Rubrik Holistik .....                   | 14          |
| 2.2.2 Rubrik Analitik .....                   | 14          |
| <b>2.3 Deep Learning.....</b>                 | <b>14</b>   |
| 2.3.1 Transformer .....                       | 15          |
| 2.3.2 Hyperparameters .....                   | 17          |
| 2.3.3 Adam Optimizer .....                    | 18          |
| <b>2.4 Natural Language Processing .....</b>  | <b>18</b>   |
| 2.4.1 Tokenisasi.....                         | 19          |
| 2.4.2 Stemming.....                           | 20          |
| 2.4.3 Stop Word Removal.....                  | 20          |
| 2.4.4 Special Character Removal.....          | 21          |
| <b>2.5 Text Similarity .....</b>              | <b>21</b>   |

|              |   |    |
|--------------|---|----|
| 2.5.1        | <i>String-based similarity</i> .....                                | 22 |
| 2.5.2        | <i>Hybrid Similarities</i> .....                                    | 22 |
| 2.5.3        | <i>Word Embeddings</i> .....  | 23 |
| 2.5.4        | <i>Cosine Similarity</i> .....                                      | 23 |
| <b>2.6</b>   | <b><i>Feedback</i></b> .....  | 24 |
| 2.6.1        | <i>Feedback Holistik</i> .....                                      | 25 |
| 2.6.2        | <i>Feedback Analitik</i> .....                                      | 25 |
| <b>2.7</b>   | <b>BERT</b> .....   | 25 |
| <b>2.8</b>   | <b><i>The Hewlett Foundation: Automated Essay Scoring</i></b> ..... | 28 |
| <b>2.9</b>   | <b><i>Cohen's Kappa Score</i></b> .....                             | 28 |
| <b>2.10</b>  | <b><i>K-Fold Cross-Validation</i></b> .....                         | 30 |
| <b>BAB 3</b> | <b>ANALISIS DAN PERANCANGAN</b> .....                               | 31 |
| <b>3.1</b>   | <b>Analisis</b> .....   | 31 |
| 3.1.1        | <i>Analisis masalah</i> .....                                       | 31 |
| 3.1.2        | <i>Analisis Data</i> .....  | 31 |
| 3.1.3        | <i>Analisis kebutuhan</i> .....                                     | 32 |
| <b>3.2</b>   | <b>Arsitektur Umum Sistem</b> .....                                 | 33 |
| <b>3.3</b>   | <b>Pengambilan dan Translasi dataset</b> .....                      | 34 |
| <b>3.4</b>   | <b><i>Pre-Processing dataset</i></b> .....                          | 35 |
| 3.4.1        | <i>Stemming</i> .....   | 35 |
| 3.4.2        | <i>Stop word removal</i> .....                                      | 36 |
| 3.4.3        | <i>Special Character Removal</i> .....                              | 38 |
| <b>3.5</b>   | <b>Representasi Vektor Word2Vec</b> .....                           | 39 |
| <b>3.6</b>   | <b>Implementasi Algoritma BERT</b> .....                            | 40 |
| <b>3.7</b>   | <b><i>Hyperparameter Tuning</i></b> .....                           | 43 |
| <b>3.8</b>   | <b>Evaluasi</b> .....   | 43 |
| 3.7.1        | <i>K-Fold Cross Validation</i> .....                                | 44 |
| 3.7.2        | <i>Cohen's Kappa Score</i> .....                                    | 44 |
| 3.7.3        | <i>Accuracy dan Loss Function</i> .....                             | 45 |
| <b>3.9</b>   | <b>Prediksi Model BERT</b> .....                                    | 45 |
| 3.8.1        | <i>BERT Score</i> .....   | 46 |
| 3.8.2        | <i>Text Similarity</i> .....  | 46 |
| 3.8.3        | <i>Rubrik Skor</i> .....  | 47 |
| 3.8.4        | <i>Feedback</i> .....   | 48 |
| <b>3.10</b>  | <b>Hasil Keluaran Sistem</b> .....                                  | 49 |

|   |    |
|---|----|
| <b>3.11 Perancangan Sistem</b>          | 49 |
| 3.10.1 <i>Use Case Diagram</i>          | 49 |
| 3.10.2 <i>Activity Diagram</i>          | 50 |
| 3.10.3 <i>Flowchart Sistem</i>          | 52 |
| <b>BAB 4 IMPLEMENTASI DAN PENGUJIAN</b> | 53 |
| <b>4.1 Implementasi Sistem</b>          | 53 |
| <b>4.2 Implementasi Website</b>         | 53 |
| <b>4.3 Pre-Processing dataset</b>       | 54 |
| <b>4.4 Representasi Vektor Word2Vec</b> | 55 |
| <b>4.5 Implementasi Algoritma BERT</b>  | 57 |
| <b>4.6 Hasil Hyperparameter Tuning</b>  | 59 |
| <b>4.7 Evaluasi</b>                     | 60 |
| <b>4.8 Prediksi Model BERT</b>          | 63 |
| 4.6.1 <i>BERT Score</i>                 | 63 |
| 4.6.2 <i>Text Similarity</i>            | 64 |
| 4.6.3 <i>Rubrik Skor</i>                | 65 |
| 4.6.4 <i>Feedback</i>                   | 67 |
| 4.6.5 <i>Final Score</i>                | 68 |
| <b>4.9 Pengujian</b>                    | 68 |
| <b>BAB 5 PENUTUP</b>                    | 83 |
| <b>5.1 Kesimpulan</b>                   | 83 |
| <b>5.2 Saran</b>                        | 83 |
| <b>DAFTAR PUSTAKA</b>                   | 85 |
| <b>Lampiran</b>                         | 88 |

## DAFTAR TABEL

|   |    |
|---|----|
| <b>Tabel 2. 1</b> Interpretasi Cohen's kappa.....                       | 29 |
| <b>Tabel 3. 1</b> Contoh stemming dataset.....                          | 35 |
| <b>Tabel 3. 2</b> Contoh <i>stop word removal</i> Dataset .....         | 37 |
| <b>Tabel 3. 3</b> Contoh <i>special character removal</i> dataset ..... | 39 |
| <b>Tabel 3. 4</b> Skenario Hyperparameter Tuning .....                  | 43 |
| <b>Tabel 4. 1</b> Hasil Hyperparameter Tuning.....                      | 60 |
| <b>Tabel 4. 2</b> Hyperparameter tuning model.....                      | 60 |
| <b>Tabel 4. 3</b> Accuracy .....  | 61 |
| <b>Tabel 4. 4</b> Kappa Score .....                                     | 62 |
| <b>Tabel 4. 5</b> Kategori Nilai .....                                  | 69 |
| <b>Tabel 4. 6</b> Pengujian Prompt Pertama .....                        | 69 |
| <b>Tabel 4. 7</b> Pengujian Prompt Kedua.....                           | 72 |
| <b>Tabel 4. 8</b> Pengujian Prompt Ketiga .....                         | 74 |
| <b>Tabel 4. 9</b> Pengujian Prompt Keempat.....                         | 77 |
| <b>Tabel 4. 10</b> Pengujian Prompt Kelima .....                        | 80 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| <b>Gambar 2. 1</b> Arsitektur Transformers .....                   | 16 |
| <b>Gambar 2. 2</b> Pre-Training dan Fine-Tuning BERT .....         | 26 |
| <b>Gambar 2. 3</b> BERT Input Representation .....                 | 27 |
| <b>Gambar 3. 1</b> Arsitektur Umum Sistem .....                    | 34 |
| <b>Gambar 3. 2</b> Proses Stemming .....                           | 35 |
| <b>Gambar 3. 3</b> Proses Remove Stop Words .....                  | 37 |
| <b>Gambar 3. 4</b> Proses Special Character Removal .....          | 38 |
| <b>Gambar 3. 5</b> Contoh Input BERT .....                         | 41 |
| <b>Gambar 3. 6</b> Contoh Output dari Inputan .....                | 41 |
| <b>Gambar 3. 7</b> Output BERT .....                               | 41 |
| <b>Gambar 3. 8</b> Use Case Diagram .....                          | 50 |
| <b>Gambar 3. 9</b> Activity Diagram .....                          | 51 |
| <b>Gambar 3. 10</b> Flowchart Sistem .....                         | 52 |
| <b>Gambar 4. 1</b> Halaman Utama .....                             | 53 |
| <b>Gambar 4. 2</b> Hasil Prediksi Esai .....                       | 54 |
| <b>Gambar 4. 3</b> Preprocess text .....                           | 54 |
| <b>Gambar 4. 4</b> Hasil Pre-processing dataset .....              | 55 |
| <b>Gambar 4. 5</b> Representasi Vektor .....                       | 56 |
| <b>Gambar 4. 6</b> Visualisasi model Word2Vec .....                | 56 |
| <b>Gambar 4. 7</b> Top 100 Vektor Word2Vec .....                   | 57 |
| <b>Gambar 4. 8</b> Pemuatan Model BERT .....                       | 58 |
| <b>Gambar 4. 9</b> Collate_fn .....                                | 58 |
| <b>Gambar 4. 10</b> Training Dataset dengan Fine-tuning BERT ..... | 59 |
| <b>Gambar 4. 11</b> Accuracy fold pertama .....                    | 61 |
| <b>Gambar 4. 12</b> Accuracy fold kedua .....                      | 61 |
| <b>Gambar 4. 13</b> Accuracy fold ketiga .....                     | 62 |
| <b>Gambar 4. 14</b> Accuracy fold keempat .....                    | 62 |
| <b>Gambar 4. 15</b> Grafik Kappa Score .....                       | 63 |
| <b>Gambar 4. 16</b> Prediksi BERT Score .....                      | 64 |
| <b>Gambar 4. 17</b> Hasil BERT Score .....                         | 64 |
| <b>Gambar 4. 18</b> Hasil Similarity .....                         | 65 |

|  |    |
|--|----|
| <b>Gambar 4. 19</b> Avg Sentence Length.....           | 65 |
| <b>Gambar 4. 20</b> Calculate Sentence Coherence ..... | 66 |
| <b>Gambar 4. 21</b> Check Main Idea.....               | 66 |
| <b>Gambar 4. 22</b> Compute Rubric Score .....         | 67 |
| <b>Gambar 4. 23</b> Hasil Rubric Score.....            | 67 |
| <b>Gambar 4. 24</b> Feedback Holistik .....            | 68 |
| <b>Gambar 4. 25</b> Feedback Analitik.....             | 68 |
| <b>Gambar 4. 26</b> Final Score.....                   | 68 |

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam perkuliahan, evaluasi pembelajaran merupakan tahapan penting terutama bagi dosen untuk melihat ketercapaian Capaian Pembelajaran Mata Kuliah (CPMK) oleh mahasiswa yang mengambil suatu mata kuliah tertentu. Salah satu bentuk evaluasi pembelajaran adalah bentuk soal esai. Soal esai memiliki kelebihan dibandingkan dengan bentuk evaluasi lainnya dimana salah satunya adalah mampu meningkatkan cara berpikir kritis atau *Higher Order Skill* (HOTS) mahasiswa. Selain itu soal esai juga memiliki kelebihan dimana dosen dapat memberikan umpan balik atau *feedback* kepada mahasiswa berdasarkan jawaban yang diberikan oleh siswa. Namun dalam prakteknya, penerapan evaluasi esai membutuhkan banyak waktu dan tenaga terutama dalam melakukan penilaian/*grading* soal esai. Salah satu solusi yaitu dengan menggunakan suatu aplikasi komputer untuk melakukan penilaian otomatis. Aplikasi ini dikenal dengan *Automated Essay Scoring* (AES) dimana dalam membangun aplikasi ini membutuhkan banyak parameter agar penilaian esai dapat mencakup berbagai aspek dan mendekati penilaian dari manusia.

Dalam proses penilaian esai, diperlukan rubrik penilaian yang digunakan untuk menilai kualitas esai berdasarkan kriteria yang diberikan oleh para dosen. Rubrik penilaian mencakup berbagai aspek, seperti pemahaman materi, pengembangan gagasan, struktur esai dan juga kesesuaian argumen. Untuk membangun AES dengan tambahan fitur *feedback*, maka AES harus dapat memberikan umpan balik berdasarkan jawaban dari mahasiswa (Kumar & Boulanger, 2021). Terdapat dua tipe rubrik dalam melakukan penilaian, yaitu rubrik analitik (*analytic rubrics*) dan rubrik holistik (*holistic rubrics*). Rubrik analitik melakukan evaluasi pada setiap kriteria secara terpisah hingga dapat memberikan *feedback* yang lebih terperinci dan fokus pada setiap aspek penilaian



yang dilakukan secara terpisah, sedangkan rubrik holistic memberikan gambaran umum tentang kualitas keseluruhan karya yang akan dinilai (Hussein et al., 2020).

Era teknologi informasi yang semakin maju, ada potensi besar untuk menggabungkan antara teknologi dengan proses penilaian esai seperti menggunakan sistem AES. AES adalah sistem penilaian esai berbasis komputer yang secara otomatis dapat memeriksa dan menilai jawaban esai mahasiswa dengan mempertimbangkan fitur-fitur yang sesuai dengan keinginan (Ramesh & Sanampudi, 2022). Hal ini bisa kita gunakan sebagai cara efektif dalam melakukan evaluasi terhadap esai. Penggunaan AES dapat meningkatkan efisiensi waktu dan skalabilitas penilaian esai secara signifikan, memungkinkan penilaian esai dilakukan kapan dan di mana saja, tidak seperti penilaian esai secara manual yang hanya dilakukan selama jam kerja dosen (Sadanand et al., 2022). Sistem ini sangat bermanfaat untuk para dosen dan juga mahasiswa yang berharap esai mereka bisa diberikan *feedback* bagian mana yang belum sesuai dan sudah sesuai.

Sistem AES umumnya terdiri dari fitur perangkat lunak. Fungsinya untuk melakukan *pre-processing* terhadap esai, mengekstraksi fitur esai yang diperlukan dan melakukan klasifikasi untuk skor akhir. Dalam melakukan *pre-processing* terhadap setiap kata, *Automated Essay Scoring* (AES) melakukan pendekatan dengan teknik *Natural Language Processing* (NLP) yang biasa digunakan dalam banyak masalah big data, ringkasan otomatis, analisis sentimen dan aplikasi lainnya (Elalfi et al., 2019).

Penelitian tentang *Automated Essay Scoring* (AES) sudah banyak dilakukan, seperti yang dilakukan oleh (Sinu, 2022) pada penelitian sistem penilaian esai secara otomatis dengan pendekatan BERT yang menghasilkan model BERT yang dilatih sudah dapat menjawab pertanyaan otomatis dengan akurasi 80%, model yang dilatih juga mendapatkan *f1-score* sebesar 69% dengan loss pada model sebesar 1.8183. Penelitian ini menggunakan *Text Similarity* tanpa melalui tokenisasi dapat menghasilkan skor yang lebih stabil dan sesuai dengan kedua teks. Model BERT yang dilatih pada penelitian ini belum begitu baik dalam menjawab pertanyaan esai yang memiliki jawaban yang panjang. Oleh karena itu, dalam penelitian yang dilakukan oleh (Sinu, 2022) dengan menggunakan *short answer grading* memiliki perbedaan dengan penelitian penilaian essay yang peneliti lakukan. Perbedaan terhadap dua penelitian ini dalam hal cakupan dan metode

penelitian yang dilakukan. Penilaian esai cenderung melibatkan analisis yang lebih mendalam dan luas terhadap pemahaman, pemikiran, dan argumentasi mahasiswa, sementara itu *short answer grading* lebih fokus pada respon yang lebih langsung dan spesifik terhadap pertanyaan.

Terdapat penelitian *Automated Essay Scoring* (AES) dengan menggunakan *Natural Language Processing* (NLP) dan *Text Mining method* yang menjelaskan bahwa sistem penilaian otomatis yang menggunakan NLP dan *Text Mining Method* mampu memberikan penilaian yang objektif, konsisten dan mendekati penilaian yang dilakukan oleh manusia. Metode ini dipengaruhi oleh beberapa proses seperti tokenisasi, *stopword*, *stemming*, jumlah kata kunci, dan sinonim kata kunci yang lebih kompleks. Penelitian ini juga menunjukkan bahwa kata kunci dan sinonim kata kunci memiliki pengaruh besar dalam penilaian otomatis esai. Sistem penilaian esai otomatis ini dapat meningkatkan efektivitas dan efisiensi dalam proses penilaian ujian online, serta memberikan hasil penilaian yang lebih cepat dan akurat daripada sistem penilaian konvensional (Gunawansyah et al., 2020).

Penelitian yang dilakukan (Y. Wang et al., 2022) yang melakukan pendekatan baru untuk *Automated Essay Scoring* (AES) menggunakan BERT. Para penulis memperkenalkan representasi esai multi-skala untuk BERT dan menggunakan *multiple losses* dan *transfer learning* untuk meningkatkan kinerja. Model ini dilatih menggunakan *multiple loss function*, termasuk *mean squared error* (MSE), *similarity* (SIM), dan *margin ranking* (MR). Hasil eksperimen menunjukkan bahwa pendekatan penelitian ini mengungguli model Deep Learning lainnya dalam tugas AES dan generalisasi dengan baik untuk tugas teks panjang. Pendekatan representasi esai multi-skala menggunakan BERT dan *multiple loss function* serta *transfer learning* yang lebih fleksibel dapat meningkatkan kinerja dalam tugas Automated Essay Scoring (AES).

Pada penelitian yang dilakukan oleh (Lahitani, 2022) dilakukan evaluasi terhadap jawaban esai berbasis teks dengan menggunakan metode cosine similarity untuk proses penilaian otomatis. Dengan pendekatan ini, hanya karakter teks yang relevan yang diberi bobot untuk menghitung kemiripan antara jawaban siswa dan kunci jawaban. Cosine similarity dianggap mewakili sekitar 52% dari nilai keseluruhan atas jawaban siswa dan terbebas dari unsur subjektif. Pengajar dapat

menambahkan poin tambahan dari aspek penilaian lain untuk memberikan keseluruhan skor yang lebih proporsional sesuai kebutuhan.

Penelitian (Lu & Cutumisu, 2021) mengimplementasikan kerangka kerja yang mengintegrasikan *Automated Essay Scoring* (AES) dengan *Automated Feedback Generation* menggunakan model Deep Learning. Penelitian ini membandingkan tiga algoritma AES dengan model *word-embedding* dan deep learning, serta mengusulkan sebuah algoritma generasi feedback baru berdasarkan *Constrained Metropolis-Hastings Sampling* (CGMH). Hasil penelitian menunjukkan bahwa algoritma AES ini memiliki performa yang lebih baik dibandingkan dengan *model-state-of-the-art* dalam hal akurasi nilai dan metode CGMH mampu menghasilkan kalimat-kalimat feedback yang terkait secara semantis. Penelitian ini menunjukkan kemungkinan untuk menggabungkan penilaian esai dengan generasi *feedback* dalam sebuah sistem otomatis. Namun, kerangka kerja ini belum diuji secara empiris dalam konteks pendidikan, dan penelitian lanjutan diperlukan untuk menggabungkan komponen linguistik ke dalam AES dan memvalidasi kerangka kerja ini.

Berdasarkan hal ini maka penelitian ini membangun sebuah sistem yang mampu mengoreksi jawaban esai bahasa Indonesia secara otomatis dan memberikan *feedback* terhadap jawaban esai. Penelitian ini menggunakan Transformer untuk melakukan pembobotan berbeda dari setiap data yang diinput. BERT (Bidirectional Encoder Representations from Transformers) merupakan model yang memodifikasi transformer, BERT dirancang untuk melatih representasi dua arah yang mendalam dari teks tanpa label dengan mengkodinisikan bersama pada konteks kiri dan kanan di semua lapisan. BERT sudah terlatih untuk disempurnakan hanya dengan satu lapisan output tambahan untuk membuat model untuk berbagai tugas, seperti menjawab pertanyaan dan inferensi bahasa tanpa memodifikasi arsitektur khusus (Devlin et al., 2019) yang diharapkan pada penelitian ini dirancang untuk dapat melakukan feedback terhadap jawaban esai mahasiswa. Dalam penelitian ini, BERT dimanfaatkan untuk menganalisis teks esai dan meningkatkan ketepatan prediksi skornya. Representasi teks yang dihasilkan oleh BERT kemudian dimasukkan ke dalam model Machine Learning yang telah dilatih untuk memperkirakan skor esai. Dengan demikian, BERT berperan dalam menginterpretasikan makna dan struktur kalimat dalam teks esai, yang berdampak

pada peningkatan akurasi prediksi skor. Sementara itu Word2Vec dimanfaatkan untuk mengukur kesamaan antara kata-kata dalam teks esai dengan kata-kata dalam referensi esai. Dengan melakukan evaluasi kesamaan ini, studi ini dapat memberikan umpan balik kualitatif yang lebih terperinci tentang kecocokan argumen, perkembangan ide, atau penerapan kata-kata kunci yang relevan.

Dataset pelatihan yang diambil merupakan dataset yang tersedia di Kaggle dari kompetisi ASAP-AES (Hamner et al., 2012). Dataset ini merupakan kumpulan data yang berharga untuk penelitian dan eksperimen dalam mengevaluasi dan meningkatkan penilaian esai berbahasa Inggris. Pada Jurnal (Chen & He, 2013) menggunakan dataset *The Hewlett Foundation: Automated Essay Scoring (ASAP)* dalam penelitiannya. Penelitian ini membagi dataset ASAP menjadi *prompt-specific rating* model dan *generic rating* model. Dataset tersebut diimplementasikan melalui pendekatan yang memanfaatkan kesepakatan antara penilaian manusia dan mesin. Hasil penelitian ini menunjukkan bahwa pendekatan yang diusulkan berhasil melampaui algoritma-algoritma terkini dan mencapai kinerja yang sebanding dengan menggunakan penilaian manusia profesional, dengan tingkat kesepakatan yang tinggi dengan penilaian manusia sekitar 0.80, diukur dengan *quadratic weighted Kappa*. Lalu, pada jurnal (Hussein et al., 2020) juga menggunakan dataset ASAP yang menunjukkan bahwa hasil penelitian yang dilakukan untuk skor keseluruhan meningkat dari 0.851 menjadi 0.85, sehingga penelitian ini memberikan kontribusi yang penting untuk sistem penilaian otomatis. Penelitian ini mengadopsi pendekatan serupa dengan jurnal (Chen & He, 2013) dan (Hussein et al., 2020), yang menggunakan dataset *The Hewlett Foundation: Automated Essay Scoring (ASAP)* sebagai dasar pembandingan, dengan tujuan untuk meningkatkan kinerja sistem penilaian otomatis esai.

Dengan akses ke dataset yang ada, studi ini melakukan analisis mendalam untuk memahami kualitas esai dengan menggunakan berbagai fitur dan karakteristik. Dataset yang berasal dari Kaggle kemudian diterjemahkan dari Bahasa Inggris ke Bahasa Indonesia untuk mendukung pengembangan model AES dalam Bahasa Indonesia. Perlu ditekankan bahwa dataset untuk Bahasa Indonesia belum banyak tersedia, sehingga langkah translasi dari dataset Bahasa Inggris menjadi kunci agar penelitian ini dapat diterapkan secara efisien dalam konteks Bahasa Indonesia. Langkah-langkah translasi dataset ini melibatkan pemetaan

struktur dan konteks Bahasa Inggris ke Bahasa Indonesia, serta memastikan konsistensi makna dan representasi dalam teks esai. Proses ini bertujuan untuk menciptakan dataset Bahasa Indonesia yang cocok untuk penelitian AES. Penelitian ini akan menyesuaikan kebutuhan dataset, library dan kategori yang akan digunakan dalam proses penelitian. Penelitian ini menghasilkan sebuah model AES dengan tambahan fitur *feedback* dimana antarmuka berupa *website* yang mudah digunakan oleh dosen maupun mahasiswa.

## 1.2 Rumusan Masalah

Soal esai merupakan sarana yang sangat signifikan dalam meningkatkan kemampuan berpikir kritis mahasiswa. Namun, menilai jawaban esai dapat menjadi sebuah tantangan bagi pendidik karena memerlukan investasi waktu dan usaha yang besar. Selain itu, memberikan umpan balik terhadap jawaban esai juga membutuhkan alokasi waktu yang cukup substansial. Untuk mengatasi hal ini, AES dengan umpan balik yang mengadopsi pendekatan BERT dan Word2Vec diusulkan. Pengembangan AES untuk Bahasa Indonesia menghadapi sejumlah kendala, termasuk keterbatasan dataset yang tersedia dan kompleksitas proses translasi. Penelitian ini diharapkan dapat menjadi solusi yang efektif dalam mengatasi tantangan dalam mengevaluasi dan memberikan respons terhadap jawaban esai mahasiswa.

## 1.3 Batasan Masalah

Beberapa batasan masalah dalam penelitian ini sebagai berikut :

1. Sistem hanya bisa memeriksa jawaban dalam bentuk teks, tidak dapat memeriksa jawaban dalam bentuk matematis.
2. Dataset yang digunakan diambil dari dataset Kaggle kompetisi ASAP-AES yang kemudian dilakukan translasi ke Bahasa Indonesia.
3. Panjang maksimum kalimat yang dapat dievaluasi oleh BERT adalah 512 kata (tokens).
4. *Feedback* yang diberikan dalam dua jenis, yaitu *feedback* analitik yang mencakup penilaian setiap komponen dan *feedback* holistik untuk penilaian keseluruhan nilai.
5. Program yang akan dirancang berbasis Website.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan sistem *Automated Essay Scoring* (AES) yang mampu memberikan penilaian otomatis terhadap esai dalam Bahasa Indonesia dengan memanfaatkan model BERT (Bidirectional Encoder Representations from Transformers). Selain itu, penelitian ini juga bertujuan untuk merancang mekanisme pemberian umpan balik (*feedback*) terhadap esai Bahasa Indonesia yang dinilai oleh sistem AES dengan menggunakan pendekatan *text similarity* berbasis Word2Vec.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu sebagai berikut

- 1) Sistem *Automated Essay Scoring* (AES) dengan pemanfaatan model BERT dan *Text Similarity* akan membantu untuk meningkatkan efisiensi penilaian esai berbahasa Indonesia. Dengan adanya sistem ini, proses penilaian akan menjadi lebih cepat. Hal ini membantu mengurangi waktu yang dibutuhkan dosen dalam mengevaluasi esai yang beragam.
- 2) Melalui pemanfaatan model BERT dan *Text Similarity*, sistem AES diharapkan mampu memberikan umpan balik atau *feedback* yang lebih cepat dan konsisten kepada mahasiswa. Hal ini akan membantu dalam mempercepat proses pembelajaran dan memberikan kesempatan kepada mahasiswa untuk dapat memperbaiki kelemahan-kelemahan yang ada pada esai mereka.
- 3) Hasil penelitian ini memiliki potensi untuk diterapkan di berbagai konteks pendidikan.

## 1.6 Metodologi Penelitian

Beberapa metode yang diterapkan dalam penelitian ini adalah sebagai berikut:

### 1. Studi Literatur

Dalam penelitian ini, peneliti melakukan observasi, mengeksplor, mempelajari dan memahami berbagai referensi yang memiliki topik sesuai dengan judul peneliti dengan tujuan untuk mengumpulkan informasi-informasi yang berkaitan dengan penelitian dan memilah data yang dibutuhkan pada penelitian

ini. Referensi yang dikumpulkan meliputi e-book, artikel-artikel ilmiah, dan penelitian yang telah dipublikasi pada jurnal yang berkaitan dengan BERT, *Automated Essay Scoring*, *Feedback*, *Text Similarity*, *Word2Vec*, *Deep Learning* dan *Natural Language Processing*.

## 2. Analisis dan Perancangan Sistem

Pada tahap analisis dilakukan analisis kebutuhan sesuai dengan tujuan penelitian yang melibatkan pemahaman yang mendalam tentang proses *Automated Essay Scoring* dan *Feedback* dengan memanfaatkan model BERT dan melakukan perancangan sistem berupa Arsitektur Umum Sistem, Perancangan Sistem dan diagram alir (*flowchart*).

## 3. Implementasi Sistem

Implementasi sistem dibangun dengan menggunakan framework streamlit sebagai sisi *front-end* untuk mengimplementasikan dalam bentuk website dan bahasa python untuk membangun model sistem.

## 4. Pengujian Sistem

Pada tahap ini akan dilakukan pengujian terhadap sistem yang telah dirancang sesuai dengan kebutuhan dan spesifikasi yang diinginkan untuk memastikan program telah dapat berjalan sesuai dengan rancangan. Sistem akan dipastikan mampu untuk melakukan proses penilaian esai bahasa indonesia dan memberikan umpan balik atau feedback terhadap jawab esai tersebut.

## 5. Dokumentasi Sistem

Tahap ini dilakukan proses dokumentasi seluruh tahap dari analisis sampai ke tahap pengujian sistem dalam bentuk skripsi untuk menunjukkan hasil dari penelitian yang telah dilakukan.

### 1.7 Penelitian Relevan

Beberapa penelitian terdahulu yang relevan dengan penelitian yang dilakukan dalam penelitian ini, antara lain:

1. Pada penelitian dengan judul *Integrating Deep Learning into An Automated Feedback Generation System for Automated Essay Scoring* (Lu & Cutumisu, 2021) mengimplementasikan kerangka kerja yang mengintegrasikan *Automated Essay Scoring* (AES) dengan *Automated Feedback Generation* menggunakan model *Deep Learning*. Penelitian ini membandingkan tiga algoritma AES dengan model *word-embedding* dan *deep learning*, serta mengusulkan sebuah algoritma generasi *feedback* baru berdasarkan *Constrained Metropolis-Hastings Sampling* (CGMH). Hasil penelitian menunjukkan bahwa algoritma AES ini memiliki performa yang lebih baik dibandingkan dengan *model-state-of-the-art* dalam hal akurasi nilai dan metode CGMH mampu menghasilkan kalimat-kalimat *feedback* yang terkait secara semantis. Penelitian ini menunjukkan kemungkinan untuk menggabungkan penilaian esai dengan generasi *feedback* dalam sebuah sistem otomatis. Namun, kerangka kerja ini belum diuji secara empiris dalam konteks pendidikan, dan penelitian lanjutan diperlukan untuk menggabungkan komponen linguistik ke dalam AES dan memvalidasi kerangka kerja ini.
2. Pada penelitian skripsi dengan judul *Sistem Penilaian Esai Secara Otomatis Untuk Open Domain Question Answering Dengan Pendekatan BERT (Bidirectional Encoder Representations From Transformers)* (Sinu, 2022) menghasilkan model BERT yang dilatih sudah dapat menjawab pertanyaan otomatis dengan akurasi 80%, model yang dilatih juga mendapatkan *f1-score* sebesar 69% dengan loss pada model sebesar 1.8183. Penelitian ini menggunakan *Text Similarity* tanpa melalui tokenisasi dapat menghasilkan skor yang lebih stabil dan sesuai dengan kedua teks. Model BERT yang dilatih pada penelitian ini belum begitu baik dalam menjawab pertanyaan esai yang memiliki jawaban yang panjang.
3. Pada penelitian dengan judul *On the Use of BERT for Automated Essay Scoring: Joint Learning of Multi-Scale Essay Representation* (Y. Wang et al., 2022) penelitian ini fokus pada penerapan model BERT untuk penilaian otomatis esai. Meskipun BERT telah dominan dalam tugas pemrosesan bahasa alami (NLP), khususnya pada *Automated Essay Scoring* (AES), penelitian menemukan bahwa pemanfaatan BERT belum sepenuhnya optimal dibandingkan dengan model *deep learning* lain seperti LSTM. Peneliti mengusulkan representasi esai multi-



skala untuk BERT yang dapat dipelajari secara bersamaan, memanfaatkan beberapa kerugian dan pembelajaran transfer dari esai di luar domain untuk meningkatkan kinerja. Eksperimen menunjukkan bahwa pendekatan ini memberikan manfaat signifikan dan mencapai hasil hampir setara dengan standar terbaik dalam tugas ASAP. Representasi esai multi-skala juga menunjukkan kinerja baik pada dataset *CommonLit Readability Prize*, menunjukkan potensi sebagai alternatif efektif untuk tugas teks panjang. Dalam konteks ini, digarisbawahi bahwa arsitektur Bi-GRU dapat mendukung pembentukan representasi esai multi-skala dengan memproses input secara *bidirectional*, menyimpan informasi dari masa lalu dan masa depan pada setiap titik waktu.

## 1.8 Sistematika Penulisan

Sistematika penulisan skripsi yang digunakan dalam penelitian ini adalah sebagai berikut:

### **BAB 1 PENDAHULUAN**

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

### **BAB 2 LANDASAN TEORI**

Bab ini menjelaskan beberapa teori yang berkaitan dengan penelitian, seperti pendahuluan *Automated Essay Scoring*, Rubrik Penilaian, *Deep Learning*, *Natural Language Processing*, *Text Similarity*, *Feedback*, *BERT*, *Dataset*, *Kappa Score* dan *K-Fold Cross-Validation*.

### **BAB 3 ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan mengenai analisis pada algoritma dan dilakukan perancangan diagram yang diperlukan, seperti diagram alir (*flowchart*).

### **BAB 4 IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi penjelasan mengenai implementasi algoritma dalam sebuah program yang kemudian diuji pada emulator mikrokontroler serta pembahasan hasil dari pengujian yang telah dilakukan.

## **BAB 5 KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan yang dapat diperoleh berdasarkan pemaparan pada setiap bab serta saran yang diberikan peneliti sebagai masukan untuk penelitian selanjutnya.

## BAB 2

### LANDASAN TEORI

#### 2.1 *Automated Essay Scoring (AES)*

##### 2.1.1 *Evaluasi Esai*

Evaluasi esai merupakan metode penilaian yang melibatkan analisis dan penilaian terhadap kualitas tulisan siswa berdasarkan berbagai kriteria yang ditetapkan oleh pengajar. Fitur-fitur evaluasi esai meliputi aspek-aspek seperti struktur tulisan, kejelasan ide, penggunaan bahasa, kohesi, koherensi, serta kemampuan analisis dan argumentasi. Dalam konteks kelas, evaluasi esai sering digunakan untuk mengukur pemahaman siswa terhadap materi pembelajaran serta kemampuan mereka dalam menyampaikan ide dan gagasan secara tertulis.

*Automated Essay Scoring (AES)* merupakan penelitian yang signifikan dalam teknologi pendidikan, dimulai sejak awal tahun 1960 dan terus berkembang seiring dengan kemajuan teknologi komputasi. AES menjadi integral dalam sistem pendidikan untuk keperluan penilaian dan aktivitas kelas di semua tingkat pendidikan. Pemanfaatan AES semakin meluas, terutama dengan adanya kegiatan kelas, penilaian, dan materi ujian yang semakin banyak tersedia secara daring. Esai yang dinilai secara otomatis dapat mencakup berbagai topik. (N. Zainal & M. H. Abu Hassan, 2022).

Penelitian terhadap penilaian teks esai secara otomatis ini umumnya terbagi menjadi dua penilaian utama, yaitu *Automated Essay Scoring (AES)* dan *Automated Short-Answer Scoring (ASAS)* yang memiliki perbedaan yang cukup signifikan. *Automated Essay Scoring* biasa digunakan untuk menilai esai yang merupakan teks yang lebih panjang dan kompleks daripada *short-answer*. Pada AES, kemampuan mahasiswa dalam menyusun argumentasi yang koheren dan mendalam merupakan hal yang menjadi tujuan utama penilaian. Untuk mengevaluasi esai, diperlukan implementasi ide-ide yang berhubungan dengan dokumen yang diminta. Sistem AES juga harus menilai kelengkapan tanggapan dan memberikan umpan balik terhadap esai (Ramesh & Sanampudi, 2022). Tidak seperti AES, *automated short-answer scoring (ASAS)* lebih menekan konten

daripada gaya penulisan, untuk menilai jawaban esai pendek dengan mengukur kesamaan jawaban esai pendek dengan model jawaban yang sesuai. Sistem yang digunakan oleh ASAS menggunakan kombinasi pendekatan sintaktik dan leksikal untuk membantu model menentukan makna semantic yang sama dalam jawaban esai singkat yang lebih sederhana (Lubis et al., 2021).

*Automated Essay Scoring* (AES) menggunakan program khusus untuk menetapkan nilai untuk esai yang ditulis dengan peraturan dalam pendidikan. Sistem AES ini dapat sangat berguna karena dapat memberikan skor esai siswa dalam hitungan detik. Hal ini merupakan bentuk penilaian pendidikan dan aplikasi *Natural Language Processing* (NLP). Tujuan utama AES adalah mengklasifikasikan sekelompok teks ke dalam beberapa kategori nilai diskrit, seperti rentang nilai 1 hingga 6. Oleh karena itu, dapat dianggap sebagai masalah klasifikasi statistik.

Beberapa faktor telah mendorong meningkatnya minat terhadap AES, termasuk kebutuhan akan penilaian yang objektif dan cepat, kesulitan dalam menilai tugas menulis secara manual, dan potensi untuk mengurangi biaya dan waktu dalam proses penilaian. Selama bertahun-tahun, sistem AES telah berkembang dengan menggunakan berbagai teknik seperti *Latent Semantic Analysis* (LSA), klasifikasi teks Bayesian, dan jaringan saraf untuk menganalisis serta memberikan nilai pada esai berdasarkan fitur-fitur linguistik dan kontekstual. Pengembangan AES telah menjadi subjek penelitian yang ekstensif, mendorong pembuatan berbagai model dan pendekatan dengan tujuan meningkatkan akurasi, keandalan, dan efisiensi dalam penilaian otomatis esai. Oleh karena itu, diperlukan penelitian lebih lanjut untuk meningkatkan efektivitas dan akurasi sistem AES di masa mendatang.

## 2.2 Rubrik Penilaian

Rubrik penilaian merupakan alat yang digunakan dalam menilai dan mengevaluasi kinerja individu atau organisasi. Umumnya, rubrik penilaian terdiri dari beberapa poin atau kriteria yang mencakup berbagai aspek tugas atau ujian. Setiap poin atau kriteria memiliki tingkatan atau kategori yang menjelaskan kemampuan kinerja individu atau organisasi. Dalam situasi ujian atau evaluasi, rubrik penilaian dapat

diterapkan untuk mengevaluasi kinerja siswa atau individu. Sebagai contoh, rubrik penilaian mungkin mencakup kriteria seperti hasil belajar, keterampilan, dan layanan. Setiap kriteria diberikan tingkatan penilaian (Hambali et al., 2022).

### 2.2.1 *Rubrik Holistik*

Rubrik holistik menilai pekerjaan secara keseluruhan dan memberikan satu skor akhir untuk tugas yang diselesaikan. Rubrik menjelaskan elemen-elemen yang harus ada dalam pekerjaan untuk memberikan panduan skor yang akan diberikan. Pada penilaian esai, rubrik holistik digunakan untuk menentukan elemen-elemen yang harus ada pada esai kemudian dilakukan penilaian secara keseluruhan dan akan diberikan *feedback* yang bersifat umum dengan mencerminkan kualitas esai mahasiswa tersebut (Badia, 2019).

### 2.2.2 *Rubrik Analitik*

Rubrik analitik memecah pekerjaan menjadi beberapa komponen yang terpisah dan memberikan skor terpisah untuk masing-masing bagian. Dalam penilaian esai rubrik holistik ini biasanya memiliki komponen seperti ide dalam esai, koherensi kalimat esai, panjang kalimat dan lain-lain. Setiap bagian komponen tersebut mendapatkan skor tertentu dan akan diberikan skor akhir sebagai total dari semua bagian. Dari setiap skor yang ada akan diberikan *feedback* yang bersifat terperinci dan menjelaskan performa setiap komponen yang ada (Badia, 2019).

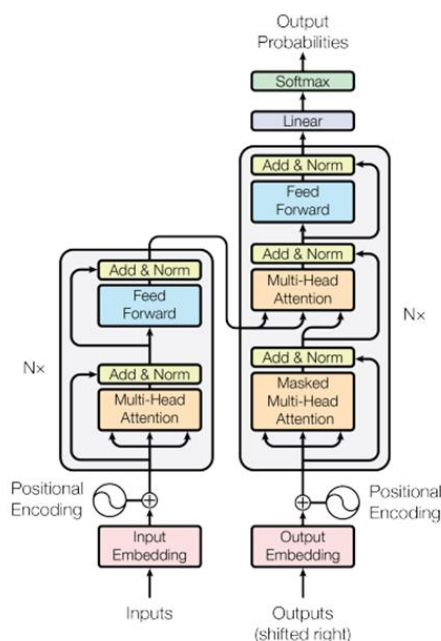
## 2.3 *Deep Learning*

*Deep Learning* merupakan sebuah subbidang pembelajaran mesin yang berfokus pada pelatihan jaringan saraf yang lebih besar pada banyak data, hal ini membutuhkan komputer yang lebih besar. Teknik yang paling populer dalam deep learning yaitu *Multilayer Perceptron Networks* (MLP), *Convolutional Neural Network* (CNN), dan *Long Short-Term Memory Recurve Neural Network* (LSTM). Ini memungkinkan penggunaan algoritma untuk memahami data dengan berbagai tingkat abstraksi. Ini dicapai dengan membangun modul-modul non-linear sederhana yang dapat mengubah representasi data dari tingkat yang lebih rendah menjadi tingkat yang lebih tinggi dan sedikit lebih abstrak.

Deep Learning juga dikenal akan skalabilitasnya, terutama dalam domain masalah di mana input dan output analog, seperti gambar, teks, atau data audio. Kelebihan ini terlihat dari kemampuan model deep learning untuk belajar tentang struktur hierarkis, sehingga membuatnya menjadi cara yang efisien untuk membangun sistem pengenalan objek. Skalabilitas pembelajaran mendalam erat hubungannya dengan ukuran data dan model yang digunakan untuk memerlukan lebih banyak daya komputasi untuk proses pelatihannya. Para ahli bidang ini, menekankan bahwa pembelajaran computer yang cukup cepat dan memiliki kelimpahan data. Oleh karena itu, deep learning bukan hanya tentang pengembangan jaringan saraf yang besar, tetapi juga tentang penerapan yang luas dan kemampuan untuk menangani berbagai tantangan dalam pemrosesan data (Brownlee Disclaimer, 2017).

### 2.3.1 *Transformer*

Transformer menjadi arsitektur yang dominan dalam deep learning. Dipadukan dengan pelatihan *self-supervised* pada dataset yang besar, *transformers* telah mencapai tingkat kemajuan terkini sehingga telah banyak diadopsi dalam *Natural Language Processing* (NLP) dan *Computer Vision* (CV). Sejak diperkenalkan pada tahun 2017, banyak muncul berbagai versi transformer yang dapat mengatasi masalah beban komputasi yang berat dan efisiensi data. Transformer lebih fleksibel dan memiliki bias yang induktif lebih sedikit dibandingkan dengan arsitektur lain seperti *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN). Transformer dikembangkan dengan menggunakan encoder dan decoder mirip dengan *Recurrent Neural Network* (RNN) sebelumnya (Schneider & Vlachos, 2023).



**Gambar 2. 1** Arsitektur *Transformers*

Arsitektur *transformers* memiliki struktur encoder dan decoder. Transformers mengikuti arsitektur umum menggunakan self-attention bertumpuk dan lapisan terhubung secara *point-wise* untuk kedua encoder dan decoder. Encoder terdiri dari tumpukan  $N=6$  lapisan identik, masing-masing memiliki dua sub-layer: mekanisme self-attention multi-head dan jaringan *feed-forward point-wise*. Decoder juga terdiri dari tumpukan  $N=6$  lapisan identik, dengan tambahan sub-layer ketiga yang melakukan multi-head attention terhadap output dari rangkaian encoder. Untuk memastikan prediksi pada suatu posisi hanya bergantung pada output yang diketahui pada posisi yang awal, sub-layer *self-attention* dalam tumpukan decoder dimodifikasi agar tidak memperhatikan posisi berikutnya. Arsitektur ini merupakan evolusi dari model berbasis jaringan saraf rekurensi atau konvolusional konvensional, karena mengandalkan sepenuhnya mekanisme *attention* dan menghapus kebutuhan rekurensi dan konvolusi secara keseluruhan. Penerapan *self-attention* memungkinkan model untuk menangkap ketergantungan antara input dan output, sehingga memungkinkan penyelarasan yang lebih efisien dan mengurangi waktu pelatihan secara signifikan (Vaswani et al., 2023).

### 2.3.2 Hyperparameters

Hyperparameter merupakan parameter-parameter yang digunakan untuk mengkonfigurasi model Machine Learning (ML) dan menentukan algoritma yang meminimalkan *loss function*. Hyperparameter ini dapat berupa *continuous*, *discrete*, *categorical*, dan *conditional hyperparameters* dengan domain dan batasan yang khusus. Optimasi hyperparameter sebagai langkah kunci dalam membangun model ML efektif, terutama untuk model kompleks seperti tree-based ML dan deep neural networks. Proses ini melibatkan pencarian ruang hyperparameter untuk mengidentifikasi kombinasi optimal. Dalam mengatasi kompleksitas, penelitian hyper-parameter optimization (HPO) terus berkembang untuk mempermudah proses penyetelan dan meningkatkan efektivitas penggunaan model ML pada masalah praktis. HPO bertujuan untuk mengotomatisasi penyetelan hyperparameter, mengurangi upaya manusia dan mencapai model ML optimal (Yang & Shami, 2022).

Hyperparameter dalam algoritma pembelajaran sebagai parameter yang mengatur proses pembelajaran dan menentukan parameter akhir model. Perubahan pada hyperparameter model machine learning dapat memiliki dampak signifikan terhadap performa prediktif model. Nilai yang tinggi pada hyperparameter “tree depth” dapat mengurangi efisiensi metode decision tree. Terdapat beberapa metode untuk menemukan hyperparameter yang optimal untuk dataset tertentu, termasuk pengaturan manual dan rekomendasi dalam paket perangkat lunak. Meskipun pengaturan default hyperparameter biasanya memberikan hasil yang memuaskan pada dataset pengujian, namun hal ini tidak selalu menghasilkan akurasi keseluruhan yang optimal (Ali et al., 2023). Dalam praktiknya, optimasi hyperparameter melibatkan pencarian ruang konfigurasi hyperparameter, metode pencarian atau optimasi dan fungsi evaluasi untuk membandingkan kinerja konfigurasi hyperparameter yang berbeda. Terdapat berbagai metode optimasi yang digunakan, seperti grid search, random search, bayesian optimization, particle swarm optimization dan genetic algorithm. Dengan demikian, hyperparameter merupakan elemen yang penting dalam proses pembangunan machine learning atau deep learning yang efisien.



### 2.3.3 *Adam Optimizer*

Adam adalah metode optimisasi stokastik yang efisien, memanfaatkan gradien orde pertama dan menghitung *learning rate* adaptif berdasarkan estimasi momen pertama dan kedua dari gradien. Nama "Adam" berasal dari estimasi momen adaptif. Adam menggabungkan keunggulan dari dua metode terkenal, yaitu AdaGrad dan RMSProp, yang efektif dalam menangani gradien yang jarang dan dalam pengaturan *online* serta non-stasioner. Salah satu keunggulannya adalah ukuran pembaruan parameter yang tetap saat gradien diubah skala, serta ukuran langkahnya dibatasi oleh *hyperparameter* langkah. Adam juga dapat beroperasi dengan baik pada objektif yang tidak stasioner dan gradien yang jarang, serta secara alami menyesuaikan ukuran langkahnya.

Adam memiliki keterkaitan yang erat dengan metode optimisasi RMSProp dan AdaGrad. RMSProp adalah metode optimisasi yang berkaitan langsung dengan Adam, dan ada versi RMSProp yang kadang-kadang menggunakan momentum. Selain itu, ada beberapa metode optimisasi stokastik lainnya seperti vSGD, AdaDelta, dan metode Newton alami yang menyesuaikan ukuran langkah dengan memperkirakan kurvatur dari informasi orde pertama. AdaMax, yang merupakan varian dari Adam, juga dibahas sebagai metode optimisasi yang berbasis pada norma tak terhingga. Adam didesain khusus untuk mengoptimalkan objektif stokastik dalam ruang parameter yang memiliki dimensi tinggi. Metode ini cocok untuk masalah dengan data besar atau banyak parameter, serta cocok untuk objektif yang tidak stasioner dan memiliki gradien yang sangat bising atau jarang. Hyperparameter Adam mudah dipahami dan biasanya memerlukan sedikit penyetelan. Selain itu, Adam telah terbukti berhasil dalam menangani berbagai masalah optimisasi *non-convex* dalam konteks *machine learning*. (Kingma & Ba, 2015)

## 2.4 *Natural Language Processing*

*Natural Language Processing* (NLP) adalah bidang kecerdasan buatan yang berfokus pada kemampuan komputer untuk memahami dan memproses bahasa manusia. NLP terdiri dari dua komponen utama, yaitu *Natural Language Understanding* (NLU) dan *Natural Language Generation* (NLG). NLP memiliki

berbagai aplikasi, seperti terjemahan mesin, deteksi spam email, ekstraksi informasi, rangkuman, dan menjawab pertanyaan. Terjemahan mesin bertujuan untuk menerjemahkan teks dari satu bahasa ke bahasa lain, sementara deteksi spam email menggunakan NLP untuk menyaring pesan yang tidak diinginkan. Metrik evaluasi NLP digunakan untuk menilai kualitas model dan sistem NLP. Beberapa metrik evaluasi yang umum digunakan meliputi tingkat kesalahan kata, skor BLEU, skor NIST, dan akurasi string generasi. Metrik-metrik ini dirancang untuk mendekati penilaian manusia tentang kefasihan dan kecukupan dalam tugas penerjemahan (Khurana et al., 2023).

Dalam memahami dan memproses bahasa manusia diperlukan beberapa teknik pra-pemrosesan data. Teknik pra-pemrosesan data memiliki peran yang krusial dalam membersihkan kata-kata, karakter, dan tanda baca, sehingga data yang diinterpretasikan lebih baik oleh mesin. Dalam *Natural Language Processing* (NLP), ada berbagai teknik pra-pemrosesan yang biasa digunakan seperti *text processing*. Penggunaan teknik *text processing* dalam *Natural Language Processing* (NLP) memiliki signifikansi besar dalam Meningkatkan kinerja algoritma. Proses ini mencakup tokenisasi, penghapusan *stopwords*, tanda baca, lemmatisasi, dan ekstraksi fitur seperti model *bag-of-words* dan TF-IDF. Proses ini sangat penting dalam menyoroti konteks kalimat atau paragraf (Tabassum & Patil, 2020).

#### 2.4.1 Tokenisasi

Tokenisasi adalah proses pemisahan sebuah teks atau kalimat menjadi unit-unit kecil yang disebut token, seperti kata-kata, karakter, atau tanda baca. Pada umumnya, kriteria pemisahan didasarkan pada keberadaan spasi atau tanda baca dalam teks. Langkah ini memiliki peran penting dalam menyaring kata-kata yang tidak diinginkan dan membantu dalam mempersiapkan data untuk langkah-langkah pemrosesan lebih lanjut (Tabassum & Patil, 2020).

Contohnya seperti :

“NLP merupakan masa depan dari sistem pengenalan ucapan!”

Kalimat tersebut akan ditokenisasi menjadi

“NLP”, “merupakan”, “masa”, “depan”, “dari”, “sistem”, “pengenalan”, “ucapan”, “!”

Dengan menerapkan tokenisasi, informasi yang terkandung dalam teks menjadi lebih terstruktur, memudahkan analisis dan ekstraksi makna dari setiap unit token yang terbentuk. Proses ini memberikan dasar bagi langkah-langkah NLP selanjutnya untuk lebih efektif memahami dan memproses data teks.

#### 2.4.2 *Stemming*

*Stemming* adalah teknik yang mencoba untuk menemukan bentuk dasar (atau akar) dari sebuah kata dengan menghapus afiksnya. Ini menggantikan kata dengan akarnya, sehingga kata-kata yang berasal dari akar yang sama dapat digabungkan (Universitas Amikom Yogyakarta et al., n.d.). Proses pemotongan akhiran, atau lemmatisasi, dilakukan untuk membawa sebuah kata ke bentuk dasarnya dengan menjatuhkan akhiran-akhiran yang mungkin melekat. Hal ini berguna untuk memperoleh kata dasar yang merepresentasikan makna utama dari kata tersebut (Tabassum & Patil, 2020). Proses ini memandang kata-kata dengan akhiran yang berbeda sebagai bentuk yang sama secara semantik. Meskipun lemmatisasi dapat meningkatkan efisiensi dan konsistensi analisis teks, terdapat situasi di mana hasilnya tidak optimal karena kata dapat kehilangan nuansa atau makna spesifik akibat pemotongan akhiran tersebut.

Contohnya seperti :

‘Antrian’ diubah menjadi ‘Antri’ dimana ‘an’ dipotong untuk mempertahankan kata dasarnya. Tetapi, ketika stemming digunakan pada kata seperti ‘Tangan’ maka hasilnya ‘Tang’ yang dimana hal itu tidak tepat.

Oleh karena itu, pemilihan antara pemotongan akhiran dan bentuk kata yang lengkap perlu disesuaikan dengan konteks dan tujuan analisis.

#### 2.4.3 *Stop Word Removal*

*Stop word removal* digunakan untuk mengeliminasi kata-kata yang kurang bermakna pada suatu dokumen dan umumnya sering muncul dalam sebuah teks (N. Chamidah et al., 2021). Contohnya, kata “yang”, “adalah”, “dan” biasanya akan dihapus pada kalimat. Hanya kata kunci yang membentuk topik yang diekstraksi. Oleh karena itu, semakin banyak StopWords yang diidentifikasi dan dihilangkan, hasil algoritma klasifikasi akan semakin baik. Penting juga untuk dicatat bahwa dalam beberapa kasus penggunaan, seperti pada model percakapan, penggunaan kata-kata negasi tertentu seperti "Tidak", "tidak bisa", "tidak akan",

"tidak" sangat penting untuk memahami konteks kalimat dan maksudnya (Tabassum & Patil, 2020).

#### 2.4.4 *Special Character Removal*

Penghapusan Karakter Khusus (Special Character Removal) adalah teknik yang sering digunakan dalam memproses teks dalam bidang pengolahan bahasa alami dan penambahan teks. Tujuannya adalah untuk membersihkan teks dari karakter-karakter yang tidak relevan atau tidak diinginkan, seperti tanda baca, simbol, angka, atau karakter non-alfanumerik lainnya. Teknik ini sangat penting karena membantu mempersiapkan teks untuk analisis atau pemrosesan lebih lanjut dengan menghilangkan gangguan yang tidak perlu. Penghapusan Karakter Khusus dapat dilakukan dengan berbagai metode, seperti menggunakan *regular expression*, substring, atau fungsi string yang tersedia dalam bahasa pemrograman. Metode ini memungkinkan penghapusan karakter secara spesifik, misalnya karakter spasi, tab, newline, atau karakter lain yang tidak diinginkan dalam suatu teks. Dengan menggunakan metode ini, teks dapat disaring secara efektif untuk memastikan hanya informasi yang relevan yang tetap ada, mempermudah analisis dan pemrosesan teks lebih lanjut.

### 2.5 *Text Similarity*

*Text Similarity* merujuk pada seberapa serupa dua teks dalam hal konten, struktur, atau makna. Berbagai metode dapat digunakan untuk mengukur tingkat kesamaan, termasuk perhitungan jarak, representasi teks, dan penerapan teknik seperti model berbasis karakter, model berbasis semantik, jaringan saraf, dan representasi berbasis grafik (J. Wang & Dong, 2020). Keberadaan *Text Similarity* memiliki peran krusial dalam konteks bahasa Indonesia untuk sejumlah tugas pemrosesan bahasa alami, seperti pencarian informasi, pengelompokan dokumen, penerjemahan mesin, dan deteksi plagiarisme. Hal ini memfasilitasi perbandingan dan kategorisasi teks, aspek yang kritis untuk mengorganisir dan mengakses informasi secara efektif. Selain itu, similarity teks memegang peran sentral dalam tugas-tugas seperti sistem rekomendasi, analisis sentimen, dan sistem tanya jawab. Keseluruhan, kemiripan teks memegang peran penting dalam memberikan kemampuan mesin untuk memahami dan memproses bahasa manusia,

menjadikannya sebagai komponen kunci dalam berbagai aplikasi pemrosesan bahasa alami (Gomaa & Fahmy, 2013).

#### 2.5.1 String-based similarity

*String-based similarity* merupakan metode pengukuran yang telah lama ada, sederhana namun banyak digunakan. Metode ini beroperasi pada urutan karakter dan komposisi karakter. Terdapat dua jenis utama fungsi kesamaan berbasis string, yakni *character-based similarity function* dan *token-based similarity functions*. *Character-based similarity* menggunakan dua string karakter dan kemudian menghitung dengan jarak edit yang termasuk penyisipan, penghapusan dan substitusi di antara keduanya. Untuk mendeteksi kesalahan dalam tata letak, *character-based measure* sangat berguna. Namun untuk mengenali istilah yang telah ditata ulang, pengukuran ini tidak bekerja dengan baik. *Term-based similarity* yang dikenal sebagai token-based dikarenakan model yang digunakan oleh string berbentuk kumpulan token-token (Prasetya et al., 2018).

#### 2.5.2 Hybrid Similarities

Selain tiga kategori metrik kesamaan yang ada, ditemukan sejumlah metrik kesamaan hibrida yang menggabungkan pendekatan berbasis string, berbasis korpus, dan berbasis pengetahuan. Contoh-contoh metrik hibrida mencakup skema Level2, SoftTFIDF, kesamaan edit umum, dan sejenisnya. Beberapa pendekatan terkini juga mencoba mengekstrak pengetahuan semantik dari WordNet dan informasi statistik di Internet. Sebagai contoh, Lin mengusulkan metode kesamaan semantik yang berbasis data terkait (LD) yang menggabungkan skema TF-IDF baru dengan metrik kesamaan kosinus yang berbasis teks klasik. Al-Hasan memperkenalkan metode kesamaan Semantik Berbasis Ontologi Inferensial (IOBSS) yang mempertimbangkan hubungan hierarki eksplisit, atribut bersama, dan hubungan implisit. Sementara itu, Atoum dan Ootom mengusulkan metode *Text Similarity Measure* (TSM) yang menggunakan informasi konten dan hubungan semantik WordNet, mencakup pencocokan kata, panjang kalimat, dan kesamaan maksimum antara kata dan teks (Prasetya et al., 2018).

### 2.5.3 Word Embeddings

Salah satu pendekatan yang populer untuk mengukur kesamaan teks adalah dengan menggunakan representasi terdistribusi kata yang dikenal sebagai *word embeddings*. *Word embeddings* adalah teknik untuk merepresentasikan kata dalam bentuk vektor numerik dalam ruang berdimensi tinggi, di mana kata-kata dengan makna yang serupa akan memiliki representasi vektor yang berdekatan. Salah satu model yang banyak digunakan untuk mendapatkan *word embeddings* adalah Word2Vec, yang diperkenalkan oleh (Mikolov et al., 2013). Word2Vec menggunakan arsitektur jaringan saraf sederhana yang dilatih untuk memprediksi kata-kata di sekitar kata target (Skip-gram) atau memprediksi kata target dari kata-kata di sekitarnya (Continuous Bag-of-Words/CBOW) (Mikolov et al., 2013).

Untuk mengukur kesamaan teks menggunakan *word embeddings*, representasi vektor dari setiap kata dalam teks digabungkan (misalnya dengan rata-rata) untuk mendapatkan representasi vektor dari keseluruhan teks. Kemudian, kesamaan antara dua vektor teks dapat dihitung menggunakan metrik kesamaan seperti *cosine similarity* atau jarak Euclidean. Pendekatan ini memiliki beberapa keunggulan dibandingkan metode tradisional seperti TF-IDF, karena dapat menangkap hubungan semantik dan sintaksis antar kata. Selain itu, *word embeddings* juga dapat mengatasi masalah kata yang jarang muncul dan kata baru yang tidak terdapat dalam kamus atau korpus teks.

### 2.5.4 Cosine Similarity

*Cosine similarity* adalah metrik yang umum digunakan untuk mengukur kesamaan dua vektor dalam ruang berdimensi tinggi. *Cosine similarity* mengukur sudut antara kedua vektor sebagai ukuran kesamaannya, bukan magnitudo. Metode ini sering digunakan dan dipelajari dalam berbagai bidang seperti penambangan data, pencarian informasi, atau penambangan teks (Halle & Young, 2001). Metrik ini mengukur kesamaan dengan menggunakan konsep orientasi antar vektor, bukan dengan membandingkan nilai absolut dari komponen vektor secara individual. Secara matematis, *cosine similarity* antara dua vektor A dan B dihitung sebagai:

$$\text{cosine\_similarity}(A, B) = (A \cdot B) / (\|A\| * \|B\|)$$

Di mana  $A \cdot B$  adalah dot product antara vektor A dan B, dan  $\|A\|$  dan  $\|B\|$  adalah panjang (norm) dari vektor A dan B. Nilai *cosine similarity* berada dalam rentang  $[-1, 1]$ , di mana nilai 1 menunjukkan kesamaan sempurna (vektor identik), nilai 0 menunjukkan tidak ada kesamaan (vektor ortogonal), dan nilai -1 menunjukkan kebalikan sempurna (vektor berlawanan arah). Dalam konteks evaluasi kesamaan teks menggunakan *word embeddings*, *cosine similarity* digunakan untuk mengukur seberapa serupa dua teks secara semantik dengan menghitung hubungan antara vektor representasi teks, seperti rata-rata vektor kata. Salah satu keunggulan utama *cosine similarity* adalah kemampuannya untuk membandingkan kesamaan teks dengan panjang vektor yang berbeda secara adil, karena tidak bergantung pada panjang vektor. Selain itu, metrik ini cenderung lebih stabil terhadap nilai-nilai ekstrem dalam vektor, sehingga menjadi lebih tangguh dalam beberapa situasi.

## 2.6 Feedback

*Feedback* atau umpan balik mencakup tindakan guru dalam membantu setiap siswa yang menghadapi kesulitan belajar secara individual dengan merespons hasil kerja siswa (Noviani et al., 2019). Umpan balik ini merujuk pada informasi yang diberikan kepada siswa setelah menyelesaikan tugas atau ujian, termasuk komentar, penilaian, atau pujian berdasarkan kinerja mereka. Pengaruh umpan balik pada kinerja siswa dapat terlihat melalui berbagai mekanisme, salah satunya adalah dampaknya pada *self-efficacy* siswa. Studi menunjukkan bahwa memberikan nilai atau skor dapat mengurangi tingkat *self-efficacy* siswa terhadap tugas atau ujian, sementara umpan balik yang bersifat deskriptif dan berfokus pada tugas dapat meningkatkan *self-efficacy* siswa. Selain itu, umpan balik deskriptif yang memberikan informasi spesifik tentang kinerja siswa dan menawarkan saran untuk mengatasi kesulitan terbukti lebih efektif dalam meningkatkan skor ujian siswa dibandingkan dengan umpan balik evaluatif yang hanya menilai seberapa baik kinerja siswa. Oleh karena itu, umpan balik kepada siswa memiliki potensi untuk memengaruhi motivasi, suasana hati, dan *self-efficacy* siswa, yang pada akhirnya akan berdampak pada kinerja siswa dalam tugas atau ujian tersebut (Lipnevich & Smith, 2009).

### 2.6.1 *Feedback Holistik*

*Feedback* holistik merupakan jenis umpan balik yang menilai karya secara keseluruhan tanpa membaginya menjadi kriteria terpisah. Dalam konteks penilaian, *feedback* ini memberikan gambaran umum tentang kualitas esai secara keseluruhan, tanpa berfokus pada detail atau aspek yang spesifik. *Feedback* holistik bertujuan untuk memberikan penilaian yang komprehensif, sehingga membantu penerima *feedback* memahami kualitas karya mereka secara umum dan mengidentifikasi area yang perlu perbaikan secara menyeluruh.

### 2.6.2 *Feedback Analitik*

*Feedback* Analitik adalah umpan balik yang memecah penilaian menjadi kriteria atau komponen yang spesifik. Dalam melakukan penilaian, umpan balik ini memberikan evaluasi yang rinci pada setiap aspek yang dinilai secara terpisah. Pendekatan ini memungkinkan pemberian umpan balik yang lebih fokus dan detail untuk setiap aspek. Dengan *feedback* analitik, penerima umpan balik mendapatkan informasi spesifik tentang kekuatan dan kelemahan mereka pada setiap kriteria, membantu mereka untuk memperbaiki kualitas karya secara mendetail.

## 2.7 BERT

Pada BERT merupakan singkatan dari *Bidirectional Representations from Transformers*. BERT merupakan model pembelajaran mesin yang dibuat untuk memproses bahasa alami secara dua arah. Model ini menggunakan dasar transformer dan dilatih tanpa memerlukan label target (Koroteev, 2021) Secara esensial, BERT merupakan serangkaian lapisan Transformer encoder yang terdiri dari berbagai '*head*' *self-attention*. Untuk setiap token input dalam urutan, setiap '*head*' menghasilkan vektor kunci, nilai, dan query, yang digunakan untuk membentuk representasi bobot. Hasil keluaran dari semua '*head*' dalam lapisan yang sama digabungkan dan dijalankan melalui lapisan yang terkoneksi sepenuhnya. Setiap lapisan dilengkapi dengan koneksi lanjutan dan diikuti oleh normalisasi lapisan. Alur kerja konvensional untuk BERT melibatkan dua tahap: *pre-training* dan *fine-tuning* (Rogers et al., 2020).

Pre-training BERT memiliki dua tahap konsep utama :



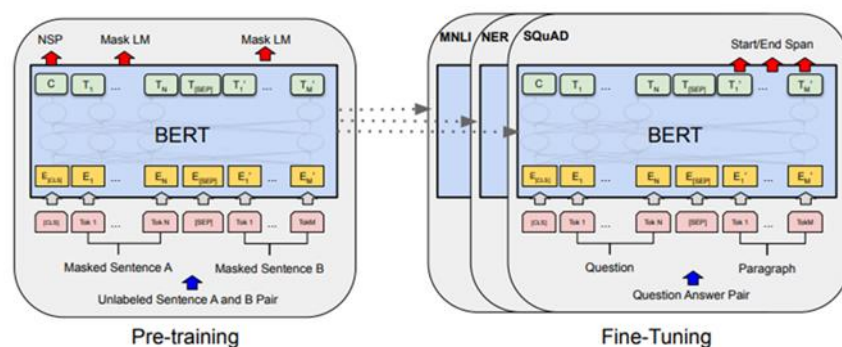
### 1. Masked Language Modelling (MLM)

BERT menerima kalimat-kalimat dengan kata random atau acak yang diisi dengan [MASK]. Model BERT mencoba untuk mengeluarkan [MASK] pada token-token. Pada tahap ini, BERT mempelajari konteks hubungan antara token kata (Djoko et al., 2020).

### 2. Next Sentence Prediction (NSP)

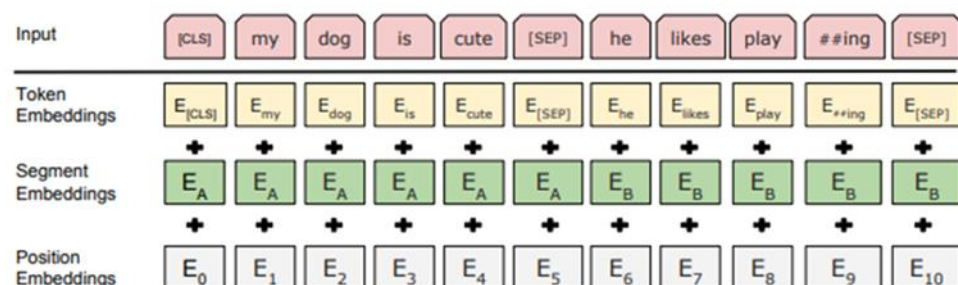
BERT mengambil 2 kalimat yang berbeda dan menentukan apakah kalimat B mengikuti makna dari kalimat A. Sehingga pada tahap ini dapat membantu model BERT untuk dapat memahami konteks dari seluruh kalimat yang berbeda (Djoko et al., 2020).

BERT menggunakan dua parameter utama dalam prosesnya, yaitu *pre-training* dan *fine-tuning*. *Pre-training* melibatkan pelatihan model tanpa label dengan tugas seperti NSP dan MLM. Sedangkan *fine-tuning* melibatkan penyesuaian model BERT yang diinisialisasi dengan parameter dari *pre-training* menggunakan data berlabel dari tugas khusus. Setelah dilakukan *pre-training*, model BERT diinisialisasi ulang pada tahapan *fine-tuning* untuk menyesuaikan dengan data yang berlabel. BERT menggunakan mekanisme *self-attention* untuk menggabungkan informasi dari kedua tahap, sehingga memungkinkan representasi teks untuk digabungkan dengan memperhitungkan hubungan antara dua kalimat. Pada tahap *fine-tuning*, penyesuaian parameter dilakukan secara *end-to-end*, menciptakan model yang efisien untuk berbagai tugas NLP (Devlin et al., 2019).



**Gambar 2. 2** Pre-Training dan Fine-Tuning BERT

Selama tahap pre-training, BERT mengubah teks input menjadi *token embeddings*, *segmentasi embeddings*, dan *positional encodings* (Djoko et al., 2020). Setelah *pre-training*, model BERT disesuaikan dengan menambahkan lapisan output agar dapat menghasilkan jawaban atas pertanyaan. Dalam *fine-tuning* seperti pada Gambar 2.2, untuk penilaian esai otomatis pada model bahasa telah diinisialisasi dengan bobot yang telah dilatih sebelumnya, dan parameter disesuaikan menggunakan dataset esai beserta skornya. Ketika menerima pertanyaan dalam bentuk token, model mencari jawaban dalam korpus dokumen menggunakan token pertanyaan dan menggunakan token khusus dalam memisahkan segmen pertanyaan dan jawaban. Ketika diberikan pertanyaan, model BERT menganalisis *embedding* yang sudah dilatih untuk mencari jawaban dan menyorot rentang teks yang mengandung jawaban. Hal ini dilakukan dengan memprediksi token yang menandai awal dan akhir dari jawaban sehingga *fine-tuning* membuat model BERT lebih spesifik untuk tugas penilaian esai (Devlin et al., 2019).



**Gambar 2. 3** BERT Input Representation

Input di BERT dibuat agar jelas merepresentasikan kalimat tunggal atau pasangan kalimat dalam satu urutan token, memungkinkan BERT mengatasi berbagai tugas *stream-down*. Representasi input pada Gambar 2.3 terbentuk dengan menjumlahkan token, *segment*, dan *position embeddings* yang sesuai untuk setiap token. Pendekatan ini memungkinkan BERT menangkap konteks dua arah dan menciptakan representasi yang bermanfaat untuk berbagai tugas. ‘Kalimat’ di sini bisa mencakup rentang teks berkelanjutan, bukan hanya kalimat linguistik. ‘Urutan’ merujuk pada urutan token input ke BERT, yang bisa satu kalimat atau dua kalimat

digabungkan. Representasi input memungkinkan BERT memproses baik kalimat tunggal maupun pasangan kalimat, seperti pasangan pertanyaan-jawaban, dalam satu urutan token terpadu, memberikan fleksibilitas untuk berbagai tugas pemrosesan bahasa alami. Secara keseluruhan, representasi input BERT didesain agar fleksibel dan dapat digunakan untuk berbagai tugas pemrosesan bahasa alami dengan kinerja tinggi.

## **2.8 *The Hewlett Foundation: Automated Essay Scoring***

Dataset *The Hewlett Foundation: Automated Essay Scoring* adalah sebuah dataset yang berisi kumpulan esai tertulis oleh siswa sekolah menengah yang digunakan untuk tugas penilaian esai otomatis (Automated Essay Scoring/AES) dalam bidang pengolahan bahasa alami (NLP). Dataset ini terdiri dari 12.976 esai yang ditulis oleh siswa kelas 8 dari delapan set tes yang berbeda. Setiap esai telah dinilai oleh setidaknya dua penilai manusia profesional menggunakan skala skor domain dari 1 hingga 6 (Keith, 2003).

Untuk menggunakan dataset ini, data esai dan skor domain yang terkait harus dimuat ke dalam program atau model AES. Data esai dapat digunakan sebagai input, sedangkan skor domain yang diberikan oleh penilai manusia dapat digunakan sebagai target atau label untuk proses pelatihan dan evaluasi model AES. Input yang diharapkan dari dataset ini adalah teks esai siswa, dan output yang diharapkan adalah skor domain yang sesuai dengan penilaian manusia. Tujuan utama dari dataset ini adalah untuk mengembangkan dan mengevaluasi sistem AES yang dapat secara otomatis memberikan skor pada esai siswa dengan akurasi yang mendekati penilaian manusia.

## **2.9 *Cohen's Kappa Score***

*Cohen's Kappa Score* adalah sebuah metrik statistik yang krusial dalam mengukur tingkat kesepakatan antara dua penilai terhadap suatu objek atau data. Metrik ini menghitung seberapa besar tingkat kesepakatan antara dua penilai dibandingkan dengan tingkat kesepakatan yang diharapkan terjadi secara kebetulan (by chance agreement), di mana nilai yang lebih tinggi menunjukkan kesepakatan yang lebih baik. Sebaliknya, nilai kappa yang rendah menunjukkan ketidaksepakatan antara penilai, sehingga membuat data menjadi kurang dapat diandalkan. Interval

kepercayaan dapat dihitung untuk statistik kappa untuk menentukan rentang kesepakatan. Penting untuk mempertimbangkan baik persentase kesepakatan maupun nilai kappa ketika menilai keandalan antar penilai. Kesepakatan antar penilai yang tinggi sangat penting dalam penelitian, terutama di industri kesehatan di mana terdapat beberapa pengumpul data yang terlibat. Nilai kappa berkisar dari -1 hingga +1, di mana nilai positif menunjukkan kesepakatan yang lebih baik daripada yang diharapkan secara kebetulan, sedangkan nilai negatif menunjukkan ketidaksepakatan yang lebih buruk daripada yang diharapkan secara kebetulan (McHugh, 2012).

**Tabel 2. 1** Interpretasi *Cohen's kappa*

| Value of Kappa | Level of Agreement | % of Data that are Reliable |
|----------------|--------------------|-----------------------------|
| 0-.20          | None               | 0-4%                        |
| .21-.39        | Minimal            | 4-15%                       |
| .40-.59        | Weak               | 15-35%                      |
| .60-.79        | Moderate           | 35-63%                      |
| .80-.90        | Strong             | 64-81%                      |
| Above .90      | Almost Perfect     | 82-100%                     |

Interpretasi Cohen's kappa yang telah ditetapkan seperti gambar diatas memuat nilai kappa  $\leq 0$  menunjukkan tidak ada kesepakatan, 0,01-0,20 menunjukkan tidak ada hingga sedikit kesepakatan, 0,21-0,49 menunjukkan kesepakatan yang cukup, 0,40-0,59 menunjukkan kesepakatan yang moderat, 0,60-0,79 menunjukkan kesepakatan yang substansial, dan 0,80-1,00 menunjukkan kesepakatan yang hampir sempurna (McHugh, 2012). Interpretasi ini membantu dalam menilai tingkat kesepakatan antara penilai berdasarkan nilai kappa yang dihasilkan. Cohen's kappa score dinilai melalui perhitungan dengan rumus sebagai berikut :

$$K = \frac{\text{Pr}(a) - \text{Pr}(e)}{1 - \text{Pr}(e)} \quad \dots(1)$$

Di mana :

K = Cohen's kappa score

$Pr(a)$  = Persentase kesepakatan yang diamati di antara penilai.

$Pr(e)$  = Persentase kesepakatan yang diharapkan secara kebetulan.

Rumus ini membantu mengukur tingkat kesepakatan antara penilai, dengan nilai kappa yang lebih tinggi menunjukkan kesepakatan yang lebih baik daripada yang diharapkan secara kebetulan.

## 2.10 K-Fold Cross-Validation

K-Fold *Cross-Validation* merupakan teknik penting dalam evaluasi model *machine learning* yang bertujuan untuk memastikan model tersebut dapat beradaptasi dengan baik pada data yang belum pernah dilihat sebelumnya. Dalam metode ini, data dibagi menjadi K bagian yang disebut *fold*. Setiap iterasi, satu *fold* digunakan sebagai data validasi sementara yang lainnya untuk melatih model. Proses ini diulang K kali dengan setiap *fold* bergantian sebagai data validasi (Mattassi, 2015).

Proses K-Fold *Cross-Validation* melibatkan beberapa langkah. Pertama, data awal diacak dan dibagi menjadi K-fold yang seukuran. Kemudian, dalam tiap iterasi, satu *fold* dipilih sebagai data validasi sementara yang lainnya digunakan untuk melatih model. Setelah itu, model dievaluasi menggunakan data validasi dan berbagai metrik seperti akurasi, presisi, *recall*, atau *F1-score* dihitung. Rata-rata dari metrik tersebut kemudian diambil untuk mengestimasi kinerja model secara keseluruhan.

Keunggulan K-Fold *Cross-Validation* antara lain memberikan estimasi yang lebih akurat tentang kinerja model pada data baru, mengurangi bias karena pemilihan data latih dan validasi yang spesifik, serta memanfaatkan semua data yang tersedia untuk pelatihan dan validasi sehingga maksimal dalam penggunaan informasi. Meskipun nilai K dapat bervariasi, nilai umumnya adalah 5 atau 10. Memilih nilai K yang lebih besar dapat memberikan estimasi yang lebih akurat namun membutuhkan waktu komputasi yang lebih tinggi.

## BAB 3

### ANALISIS DAN PERANCANGAN

#### 3.1 Analisis

Analisis adalah tahap awal dalam memahami suatu informasi, yang melibatkan pemecahan dan pemilahan informasi menjadi bagian-bagian yang lebih kecil. Tujuannya adalah agar informasi tersebut menjadi lebih mudah dipahami. Proses analisis merupakan langkah penting sebelum melakukan perancangan dan pengembangan sistem tertentu. Hal ini dilakukan agar sistem yang dikembangkan dapat memenuhi kebutuhan yang ada dan menjadi lebih terstruktur dalam mencapai tujuan akhirnya.

##### 3.1.1 Analisis masalah

Dalam konteks penilaian esai dalam bahasa Indonesia, beberapa tantangan mendasar perlu diatasi. Pertama, metode penilaian manual seringkali memakan waktu dan rentan terhadap subjektivitas penilai, menghasilkan ketidakpastian dan kurangnya konsistensi dalam penilaian. Selain itu, proses manual tidak praktis untuk volume besar esai dan tidak memberikan umpan balik yang cepat dan berkualitas kepada siswa. Meskipun teknologi penilaian otomatis telah berkembang pesat, keterbatasan dalam bahasa Indonesia masih menjadi masalah, mengingat keberagaman struktur esai dan representasi teks yang kompleks. Integrasi model BERT dan *text similarity* berbasis Word2Vec menawarkan potensi solusi, tetapi masih memerlukan pemahaman mendalam tentang bahasa dan validitas penilaian yang sesuai. Oleh karena itu, penelitian ini menjadi penting untuk mengatasi tantangan ini dan meningkatkan efektivitas penilaian otomatis dan pemberian umpan balik pada esai dalam bahasa Indonesia.

##### 3.1.2 Analisis Data

Penelitian ini menggunakan *dataset* yang diambil dari website Kaggle dengan *dataset* berjudul *The Hewlett Foundation: Automated Essay Scoring* yang berisi informasi tentang skor esai yang dinilai secara otomatis menggunakan sistem penilaian esai otomatis. Set data yang terdapat *dataset* ini dapat digunakan untuk menganalisis performa dan kinerja dari system AES dalam menilai esai siswa.

*Dataset* ini berisi informasi tentang skor esai yang diberikan oleh sistem AES dan oleh dua penilaian manusia. Data ini terdiri dari delapan set esai yang berbeda, dengan total 12.976 esai. Setiap esai dinilai dalam kisaran skor 0 hingga 6. Pada penelitian ini, data yang digunakan hanya mencakup dua set esai yang terdiri dari 3.584 esai yang memiliki *domain1\_score* sebagai nilai pembanding dengan nilai sistem yang akan dibuat. Pada tahapan ini *dataset* digunakan untuk melakukan proses mendapatkan nilai dengan melakukan analisis tingkat kesepakatan antara skor yang diberikan oleh dua penilaian dengan menggunakan Kappa score sehingga dapat dilakukan analisis hubungan antara fitur teks esai dengan skor yang diberikan.

### 3.1.3 Analisis kebutuhan

Analisis kebutuhan adalah langkah krusial yang bertujuan untuk mengidentifikasi dan memahami kebutuhan yang diperlukan agar sistem yang dirancang dapat mencapai tujuan yang ditetapkan. Proses ini terbagi menjadi dua aspek utama, yaitu fungsional dan non-fungsional.

#### 1. Kebutuhan fungsional

Kebutuhan fungsional adalah kumpulan fitur atau fungsi yang harus ada dalam suatu sistem atau produk untuk memenuhi kebutuhan pengguna atau pemangku kepentingan. Ini mencakup spesifikasi fungsionalitas yang harus ada pada sistem untuk mencapai tujuan utama. Penelitian ini memiliki kebutuhan fungsional utama, yaitu:

- a. Menerima esai Bahasa Indonesia sebagai inputan penilaian.
- b. Melakukan proses *pre-processing* dataset seperti *stemming*, *stop word removal* dan *special character removal* untuk menghasilkan dataset yang lebih dipahami oleh sistem.
- c. Menggunakan BERT sebagai model untuk klasifikasi urutan untuk menilai atau mengklasifikasikan esai berdasarkan nilai yang diberikan.
- d. Hasil dari BERT digunakan untuk proses penilaian esai dengan membandingkan antara esai dengan referensi esai menggunakan *Text Similarity*.
- e. Memberikan rubrik skor dengan penilaian terhadap panjang kalimat, koherensi kalimat dan ide utama pada esai.

- f. Proses penilaian nilai akhir esai dilakukan dengan mengkombinasikan hasil nilai dari BERT, nilai *text similarity*, dan nilai dari rubrik skor.
- g. *Feedback* diberikan dengan menyusun umpan balik sesuai dengan hasil akhir dari penilaian nilai esai.
- h. Menghasilkan keluaran berupa hasil dari esai dan umpan balik terhadap esai.

## 2. Kebutuhan non-fungsional

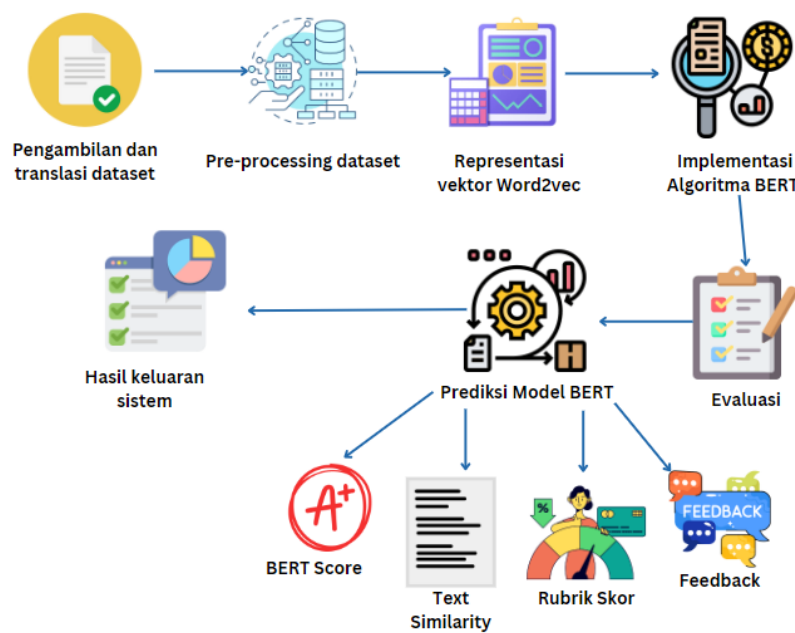
Kebutuhan non-fungsional adalah spesifikasi tambahan yang mendukung sistem, dapat berupa kinerja, keamanan, serta batasan dari sistem. Penelitian ini memiliki beberapa kebutuhan non-fungsional, yaitu:

- a. Sistem memiliki tampilan yang *user friendly*.
- b. Memiliki fitur untuk input jawaban esai yang akan diberikan skor ketika melakukan klik tombol *get score*.
- c. Sistem dapat menerima inputan dan proses data dalam waktu yang singkat dan hasil yang akurat.

## 3.2 Arsitektur Umum Sistem

Arsitektur umum sistem yaitu melakukan pengambilan dataset pada Kaggle kemudian dilakukan proses translasi dataset ke Bahasa Indonesia. Kemudian data yang sudah ada dilakukan pengolahan melalui proses *pre-processing* dataset yang meliputi *stemming*, *stop word removal* dan *remove special character*. Lalu setelah dataset diolah, kemudian dilakukan proses pelatihan model Word2Vec untuk menghasil representasi vektor untuk setiap kata pada dataset. Kemudian dilakukan implementasi Algoritma BERT dengan cara melatih model untuk melakukan klasifikasi urutan untuk nilai sehingga akan dilakukan evaluasi untuk penilaian esai. Setelah melakukan evaluasi, model BERT yang sudah dilatih kemudian dilakukan percobaan prediksi nilai esai dengan menggabungkan penilaian *text similarity*, pemberian rubrik skor dan *feedback* terhadap esai. Lalu, hasil keluaran sistem berupa nilai dan *feedback*.





**Gambar 3. 1** Arsitektur Umum Sistem

### 3.3 Pengambilan dan Translasi dataset

Dalam memulai pengembangan sistem *Automated Essay Scoring*, langkah awalnya adalah mengambil dataset yang relevan. Salah satu sumber dataset yang populer adalah Kaggle, sebuah platform yang menyediakan beragam dataset untuk berbagai keperluan. Dalam konteks ini, dataset yang diambil berasal dari Kaggle dengan judul "*The Hewlett Foundation: Automated Essay Scoring*". Setelah dataset diunduh, langkah selanjutnya adalah memilih data untuk memilih subset yang sesuai dengan kebutuhan proyek. Dari 12.967 dataset yang tersedia, dipilih 3.684 dataset yang dianggap representatif dan relevan untuk digunakan dalam pengembangan sistem. Selanjutnya, dataset tersebut menjalani proses translasi atau terjemahan dari Bahasa Inggris ke Bahasa Indonesia. Proses translasi ini penting untuk mendukung pengembangan sistem yang mempertimbangkan konteks Bahasa Indonesia, sehingga memungkinkan sistem untuk melakukan penilaian otomatis terhadap esai dalam Bahasa Indonesia dengan lebih akurat dan efektif.

### 3.4 Pre-Processing dataset

*Pre-processing* dataset adalah serangkaian langkah yang dilakukan untuk membersihkan, mengubah, dan mempersiapkan data mentah menjadi bentuk yang lebih sesuai untuk analisis atau pemodelan. Tujuannya adalah untuk meningkatkan kualitas data, menghilangkan noise atau ketidaksesuaian, dan membuatnya siap untuk dieksploitasi oleh algoritma pemrosesan data atau pembelajaran mesin. Tahapan yang dilakukan pada *pre-processing* penelitian ini adalah *stemming*, *stop word removal* dan *special character removal*.

#### 3.4.1 Stemming

Stemming merupakan tahapan *pre-processing* yang bertujuan untuk mengubah kata-kata menjadi bentuk dasarnya atau kata dasar. Hal ini berguna untuk memperoleh kata dasar yang merepresentasikan makna utama dari kata tersebut (Tabassum & Patil, 2020).

```
[ ] def stem_text(text):
    stemmer = PorterStemmer()
    tokens = tokenize(text)
    stemmed_tokens = [stemmer.stem(token) for token in tokens]
    return ' '.join(stemmed_tokens)
```

**Gambar 3. 2** Proses *Stemming*

Dengan menggunakan library NLTK dari Python, fungsi yang digunakan untuk melakukan stemming adalah menerima teks sebagai input dan mengembalikan teks yang telah di stemming. Contoh dari *stemming* sebagaimana terlihat pada Tabel 3.1.

**Tabel 3. 1** Contoh *stemming* dataset

| Esai   | Hasil <i>stemming</i>  |
|--|--|
| Kecerdasan Buatan (AI) adalah teknologi yang memungkinkan mesin untuk meniru dan melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia, seperti belajar, merencanakan, dan memecahkan masalah. AI telah membawa dampak | Kecerdasan Buatan (AI) adalah teknologi yang mungkin mesin untuk tiru dan laku tugas-tugas yang biasa perlu cerdas manusia, seperti ajar, rencana, dan pecah masalah. AI telah bawa dampak besar dalam berbagai bidang, mulai dari produksi industri |

|  |  |
|--|--|
| besar dalam berbagai bidang, mulai dari produksi industri hingga perawatan kesehatan, dan telah mengubah cara kita berinteraksi dengan teknologi. Meskipun memiliki potensi besar untuk meningkatkan efisiensi dan inovasi, penggunaan AI juga memunculkan berbagai isu etika, sosial, dan ekonomi yang perlu dipertimbangkan dengan cermat. | hingga rawat kesehatan, dan telah ubah cara kita interaksi dengan teknologi. Meski miliki potensi besar untuk tingkat efisiensi dan inovasi, guna AI juga muncul berbagai isu etika, sosial, dan ekonomi yang perlu timbang dengan cermat. |
| Bermain game secara berlebihan dapat menyebabkan dampak negatif pada kesehatan fisik dan mental, seperti kurangnya aktivitas fisik, gangguan tidur, peningkatan risiko obesitas, serta masalah psikologis seperti kecanduan dan isolasi sosial.  | Bermain game secara berlebih dapat sebab dampak negatif pada sehat fisik dan mental, seperti kurang aktivitas fisik, ganggu tidur, tingkat risiko obesitas, serta masalah psikologis seperti candu dan isolasi sosial.                     |
| Pengaruh komputer meliputi kemudahan akses informasi, peningkatan efisiensi dalam pekerjaan, serta meningkatnya ketergantungan pada teknologi dan potensi dampak negatif seperti isolasi sosial dan gangguan tidur.  | Pengaruh komputer meliputi mudah akses informasi, tingkat efisiensi dalam kerja, serta tingkat ketergantungan pada teknologi dan potensi dampak negatif seperti isolasi sosial dan ganggu tidur.   |

### 3.4.2 Stop word removal

*Stop word removal* adalah proses dalam pemrosesan teks yang bertujuan untuk menghapus kata-kata yang umumnya tidak memberikan nilai tambah dalam pemahaman konten teks. Kata-kata ini sering muncul dalam teks tetapi tidak membawa makna yang signifikan dalam analisis teks atau pemodelan bahasa.

```

def remove_stop_words(text):
    stop_words = set(stopwords.words('indonesian'))
    tokens = tokenize(text)
    filtered_tokens = [token for token in tokens if token.lower() not in stop_words]
    return ' '.join(filtered_tokens)

```

**Gambar 3. 3** Proses *Remove Stop Words*

Dalam gambar 3.3 merupakan proses penghapusan *stop words* dari sebuah teks dalam bahasa Indonesia. Dalam tahap ini, fungsi mengimpor daftar *stop words* bahasa Indonesia dari library 'nltk' lalu kemudian dilakukan proses pemisahan menjadi token-token yang akan difilter untuk memeriksa apakah setiap token tidak ada dalam daftar *stop words*. Token-token yang lolos filter akan dipertahankan, sementara token yang merupakan *stop words* akan dihapus. Kemudian token-token yang sudah difilter akan digabungkan menjadi satu teks yang telah dibersihkan dan digunakan dalam analisis teks selanjutnya. Agar memperjelas tahapan *stop word removal* contoh pada tahap ini dapat dilihat pada tabel 3.2.

**Tabel 3. 2** Contoh *stop word removal* Dataset

| Esai   | Hasil <i>stop word removal</i>  |
|--|---|
| Bermain game <i>secara</i> berlebihan <i>dapat</i> menyebabkan dampak negatif pada kesehatan fisik <i>dan</i> mental, <i>seperti</i> <i>kurangnya</i> aktivitas fisik, gangguan tidur, <i>peningkatan</i> risiko obesitas, <i>serta</i> masalah psikologis <i>seperti</i> kecanduan <i>dan</i> isolasi sosial. | Bermain game berlebihan menyebabkan dampak negatif kesehatan fisik mental, aktivitas fisik, gangguan tidur, risiko obesitas, masalah psikologis kecanduan isolasi sosial. |
| Pengaruh komputer <i>meliputi</i> kemudahan akses informasi, peningkatan efisiensi <i>dalam</i> pekerjaan, <i>serta</i> meningkatnya ketergantungan <i>pada</i> teknologi <i>dan</i> potensi dampak negatif <i>seperti</i> isolasi sosial <i>dan</i> gangguan tidur.   | Pengaruh komputer kemudahan akses informasi, peningkatan efisiensi pekerjaan, meningkatnya ketergantungan teknologi potensi dampak negatif isolasi sosial gangguan tidur. |

Kata yang diwarnai biru merupakan kata yang dianggap umum sehingga dilakukan penghapusan dalam kalimat, terkadang dengan dilakukan proses *stop word removal* ini ada kata yang penting dalam koherensi antara kalimat sehingga harus lebih diperhatikan lagi kegunaannya.

### 3.4.3 Special Character Removal

*Special character removal* adalah proses dalam pemrosesan teks yang bertujuan untuk menghapus karakter khusus atau simbol yang tidak diperlukan dari teks. Karakter khusus ini dapat termasuk tanda baca seperti titik, koma, tanda seru, tanda tanya, dan lainnya, serta karakter khusus seperti simbol matematika, emotikon, dan karakter non-alfanumerik lainnya.

```
[ ] def remove_special_chars(text):
    # Menghapus karakter tab, baris baru, back slice, dan karakter non-ASCII
    text = re.sub(r'[\t\n\r\x0c\x0b]', ' ', text)
    text = re.sub(r'^[\x00-\x7F]+', '', text)

    # Mengubah teks menjadi huruf kecil
    text = text.lower()

    # Menghapus tanda baca
    text = re.sub(r'[\W\s]', '', text)

    # Menghapus mention, link, hashtag, dan URL yang tidak lengkap
    text = re.sub(r'(@\w+)|((https?://|www\.)?\w+\.\w+)', '', text)

    return text
```

**Gambar 3. 4** Proses *Special Character Removal*

Gambar 3.4 merupakan *pseudocode* yang digunakan dalam melakukan penghapusan karakter khusus. Dalam proses *special character removal* digunakan *regular expression* (regex) dalam *python*. Regex digunakan untuk melakukan operasi pencarian, penggantian, dan pengolahan teks pada string. Terdapat beberapa fungsi ‘re.sub’ yang menggunakan regex untuk melakukan operasi penghapusan atau penggantian pada teks yang diberikan.

1. “re.sub(r'[\t\n\r\x0c\x0b]', ' ', text)” digunakan untuk mengganti karakter tab, baris baru, *carriage return*, *form feed*, dan *vertical tab* dengan spasi.
2. “re.sub(r'^[\x00-\x7F]+', '', text)” digunakan untuk menghapus karakter non-ASCII.
3. “text.lower()” digunakan untuk mengubah semua karakter dalam teks menjadi huruf kecil.

4. “re.sub(r'[^\w\s\']', '', text)” menghapus semua tanda baca.
5. “re.sub(r'(@\w+)|((https?://www\|.)\w+\.\w+)', '', text)” digunakan sebagai menghapus *mention*, link, dan hastag.

Contoh tahapan *character removal* dapat dilihat dari tabel 3.3.

**Tabel 3. 3** Contoh *special character removal* dataset

| Esai  | Hasil <i>special character removal</i>  |
|---|---|
| Pengaruh bermain game sangatlah signifikan!!! Dampaknya mencakup peningkatan keterampilan kognitif dan motorik, kepuasan hiburan, serta potensi risiko kecanduan dan gangguan kesehatan mental :( . Jangan lupakan pentingnya pengawasan dan batasan waktu! #GameOn | Pengaruh bermain game sangatlah signifikan Dampaknya mencakup peningkatan keterampilan kognitif dan motorik, kepuasan hiburan, serta potensi risiko kecanduan dan gangguan kesehatan mental Jangan lupakan pentingnya pengawasan dan batasan waktu GameOn |

### 3.5 Representasi Vektor *Word2Vec*

Setelah melakukan tahapan *pre-processing*, selanjutnya dilakukan representasi vektor yang biasa dikenal sebagai embedding. Pada tahapan ini, Model Word2Vec dilatih terlebih dahulu menggunakan daftar kata dari data pelatihan sehingga akan menghasilkan vektor untuk setiap kata dalam kosakata. Word2Vec memanfaatkan konteks kata dalam suatu korpus teks untuk mempelajari representasi vektor kata yang dapat menangkap hubungan semantik antara kata-kata.

Representasi Vektor Word2Vec yang diimplementasikan menggunakan arsitektur Skip-gram yang bertujuan untuk memprediksi konteks kata-kata berdasarkan kata target. Misalnya, jika kata target adalah “kucing”, model akan mempelajari representasi vektor dari kata “kucing” untuk memprediksi konteks kata-kata seperti “hewan” dan “peliharaan”. Vektor-vektor kata ini memiliki properti yang menarik, yaitu kata-kata yang memiliki makna atau konteks yang serupa akan memiliki vektor yang mirip dalam ruang vektor multidimensi. Ini berarti operasi aljabar seperti penjumlahan atau pengurangan vektor kata dapat digunakan untuk mengeksplorasi hubungan semantik antara kata-kata. Setelah

melakukan pelatihan model untuk representasi vektor, maka selanjutnya dilakukan vektor fitur untuk setiap esai yang dibuat dengan mengambil rata-rata vektor kata yang menyusun esai tersebut. Representasi vektor tersebut kemudian akan divisualisasikan dalam dimensi yang lebih rendah untuk memahami hubungan semantic antara kata-kata.

### 3.6 Implementasi Algoritma BERT

Setelah melakukan representasi vektor kata menggunakan Word2Vec, tahap selanjutnya adalah membangun model berdasarkan dataset yang telah diproses menggunakan algoritma BERT. Pada penelitian ini, model BERT akan diimplementasikan untuk memprediksi skor esai dengan memanfaatkan metrik evaluasi *Cohen's Kappa Score*. Penelitian ini akan menggunakan *tokenizer* dan model BERT dari *library transformers Hugging Face*. Parameter-parameter seperti laju pembelajaran (*learning rate*), ukuran batch (*batch size*), dan jumlah *epoch* memiliki peran penting dalam proses pelatihan model menggunakan algoritma BERT. Laju pembelajaran mengontrol seberapa besar perubahan bobot model setiap kali dilakukan pembaruan, yang mempengaruhi kecepatan konvergensi model. Jika laju pembelajaran terlalu besar, model mungkin sulit mencapai minimum global, sedangkan nilai yang terlalu kecil dapat memperlambat proses pelatihan. Di sisi lain, ukuran batch menentukan jumlah sampel yang diproses secara bersamaan dalam satu iterasi. Ukuran batch yang besar dapat mempercepat pelatihan dengan memanfaatkan paralelisme, tetapi juga membutuhkan lebih banyak memori GPU. Selanjutnya, jumlah epoch mengatur berapa kali seluruh dataset akan diberikan ke model. Jumlah epoch yang tepat diperlukan agar model memiliki cukup waktu untuk mempelajari pola yang kompleks tanpa *overfitting* pada data pelatihan. Dengan mengatur parameter-parameter ini secara optimal, kinerja dan efisiensi model BERT dalam melakukan penilaian otomatis terhadap esai dapat ditingkatkan.

Dalam proses penilaian menggunakan model BERT, data esai akan diproses dengan memuat teks esai dan memisahkannya dari skor yang sesuai. Teks esai akan ditokenisasi menggunakan *tokenizer* BERT untuk menghasilkan representasi token, *attention mask*, dan label skor sebenarnya. Contoh dari input yang akan

menghasilkan output untuk digunakan pada proses selanjutnya seperti pada gambar 3.2 ini.

```
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
text = "Machine learning is a subfield of artificial intelligence."
inputs = tokenizer(text, return_tensors='pt', padding=True, truncation=True)
print(inputs)
```

**Gambar 3. 5** Contoh *Input* BERT

Pada tahap ini, input yang dimasukkan berupa text yang kemudian akan dilakukan *tokenize* untuk dapat menghasilkan output berupa '*input\_ids*', '*token\_type\_ids*', dan '*attention\_mask*' yang dapat kita lihat contoh outputnya pada gambar 3.3.

```
{'input_ids': tensor([[ 101, 7845, 2025, 1996, 2117, 3884, 1010, 2292, 2116, 102]]),
 'token_type_ids': tensor([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]),
 'attention_mask': tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])}
```

**Gambar 3. 6** Contoh *Output* dari Inputan

Data yang telah diproses akan diubah menjadi *DataLoader* untuk mengemas data esai dan label ke dalam format yang dapat diterima oleh model BERT. Model BERT yang digunakan adalah '*BertForSequenceClassification*', yang merupakan model BERT yang dilatih untuk melakukan tugas klasifikasi urutan.

```
tensor([[0.9070, 0.0930]], grad_fn=<SoftmaxBackward0>)
```

**Gambar 3. 7** *Output* BERT

Hasil dari data yang diproses oleh model BERT dapat kita lihat pada gambar 3.4 yang dimana hasilnya berupa teks yang telah diubah menjadi representasi numerik yang kemudian akan digunakan untuk melakukan klasifikasi teks hingga penilaian.

Model ini menerima input berupa teks esai dan skor yang sesuai. Tahap pelatihan dilakukan dengan mengoptimalkan parameter model menggunakan *optimizer Adam* dan *loss function* yang sesuai. Setelah memuat model BERT, tahap selanjutnya adalah mempersiapkan Data Pelatihan dan Pengujian program dengan menggunakan teknik *Cross-Validation K-Fold* untuk membagi data menjadi



kelompok pelatihan dan pengujian. Data esai diambil dari file esai yang telah diproses sebelumnya dan dibagi menjadi kelompok-kelompok dengan menggunakan *KFold* dari *scikit-learn*.

Setelah mengemas data esai dan mempersiapkan model untuk pelatihan, proses selanjutnya adalah melatih model menggunakan algoritma BERT. Pada setiap *epoch* pelatihan, seluruh data pelatihan akan diproses dalam batch-batch kecil. Masukan dari setiap batch adalah teks esai dataset yang diberi label '*essays*', sedangkan label yang sesuai adalah skor esai yang telah diketahui dari dataset yang dimana diberi label '*domain1\_score*'. Model BERT kemudian menerima masukan ini dan melakukan prediksi terhadap skor esai (*y\_pred*) berdasarkan teks esai. Selanjutnya, kerugian (*loss*) dihitung dengan membandingkan prediksi (*y\_pred*) dengan label sebenarnya (*y\_true*), yang merupakan skor esai dalam data pelatihan. Gradien dari kerugian ini kemudian dihitung menggunakan algoritma *backpropagation*, dan bobot model diperbarui menggunakan *optimizer Adam*.

Dalam proses klasifikasi teks esai, model BERT menggunakan representasi token dan *attention mask* untuk memproses teks. Representasi token digunakan untuk memahami makna teks, sedangkan mask perhatian (*attention mask*) membantu model untuk fokus pada bagian-bagian penting. Dengan cara ini, model BERT dapat menghasilkan prediksi skor esai yang sesuai dengan teks yang diberikan. Dalam konteks penilaian, output dari model BERT adalah prediksi skor esai berdasarkan teks yang diberikan. Artinya, sistem akan menerima teks esai sebagai masukan, dan kemudian mengeluarkan prediksi berupa skor numerik yang merepresentasikan penilaian terhadap teks tersebut. Dengan pendekatan ini, sistem penilaian dapat memberikan penilaian otomatis terhadap esai menggunakan model BERT. Selama proses pelatihan, grafik kerugian dan akurasi dipantau untuk setiap lipatan data. Setelah pelatihan selesai, model dievaluasi menggunakan data pengujian untuk menghitung *Cohen's Kappa Score* dan akurasi. Hasil prediksi model dibandingkan dengan label sebenarnya untuk mengevaluasi kinerja model. Prediksi yang dihasilkan oleh model disimpan dalam file CSV, sedangkan model yang telah dilatih disimpan ke dalam file untuk penggunaan selanjutnya. Dengan demikian, proses penilaian menggunakan algoritma BERT memberikan sistem

penilaian yang andal dan efektif, memungkinkan penilaian otomatis terhadap esai dengan tingkat akurasi yang tinggi.

### 3.7 Hyperparameter Tuning

Perencanaan *hyperparameter tuning* merupakan langkah krusial dalam pengembangan model *machine learning* atau *deep learning*. *Hyperparameter* adalah parameter yang nilainya harus ditentukan sebelum proses pelatihan, seperti *learning rate*, jumlah *epoch*, ukuran batch, dan arsitektur jaringan. Penentuan nilai *hyperparameter* yang optimal sangat penting karena dapat mempengaruhi kinerja model secara signifikan. Skenario untuk *hyperparameter tuning* pada sistem ini dapat dilihat pada tabel 3.4 ini.

**Tabel 3. 4** Skenario *Hyperparameter Tuning*

| Epoch | Batch Size | Learning Rate | K-Fold Cross-validation |
|-------|------------|---------------|-------------------------|
| 20    | 8          | 2e-5          | 3                       |
| 30    | 8          | 3e-5          | 3                       |
| 30    | 8          | 1e-5          | 4                       |

Dalam perencanaan *hyperparameter tuning*, langkah utama adalah mengidentifikasi *hyperparameter* yang paling berpengaruh, menentukan rentang nilai yang mungkin, serta memilih metode yang akan digunakan seperti *grid search* atau *random search*. Validasi silang juga sangat penting untuk menghindari *overfitting* pada data pelatihan. Dengan perencanaan yang matang, sumber daya komputasi dapat digunakan secara efisien dan efektif, sehingga diperoleh model dengan kinerja optimal dalam waktu yang wajar.

### 3.8 Evaluasi

Setelah algoritma BERT diimplementasikan, akan dilakukan tahapan evaluasi untuk mengetahui keakuratan proses pelatihan BERT untuk melakukan tugas klasifikasi sekuens dan kemudian akan dihitung skor Kappa antara label prediksi dan label yang sebenarnya untuk mendapatkan prediksi skor esai tersebut dengan

menggunakan `cohen_kappa_score` dari *scikit-learn*. Dalam hal ini, akan dilakukan beberapa jenis evaluasi yaitu :

### 3.7.1 K-Fold Cross Validation

K-Fold *Cross Validation* digunakan dalam proses evaluasi pada program karena dengan menggunakan metode ini, program dapat memberikan validasi model yang lebih baik, mengurangi *overfitting*, memanfaatkan dataset yang terbatas, membandingkan model dengan adil dan menghitung metrik evaluasi yang lebih stabil. Akan dilakukan *cross validation* dengan jumlah *fold* sebanyak 4. Data dibagi menjadi data train dan data test secara bergantian, sehingga model BERT dilatih dan dievaluasi pada setiap *fold*, dan rata-rata metrik akan dihitung dari semua *fold*.

### 3.7.2 Cohen's Kappa Score

Evaluasi menggunakan *Cohen's Kappa Score* dilakukan dalam penelitian ini karena beberapa pertimbangan penting. Pertama, penelitian ini berurusan dengan masalah klasifikasi multilabel, di mana setiap sampel data dapat diklasifikasikan ke dalam beberapa kategori atau skor. Dalam konteks ini, penggunaan *Cohen's Kappa Score* lebih disukai daripada hanya menggunakan metrik evaluasi sederhana, karena mampu menangani masalah multilabel dengan lebih baik. Kedua, *Cohen's Kappa Score* dikembangkan untuk mengukur tingkat kesepakatan antara dua penilai atau anotator dalam memberikan penilaian atau label terhadap suatu set data. Dalam penelitian ini, model BERT yang digunakan dapat dipandang sebagai satu “anotator” yang memberikan prediksi kelas, sedangkan label sebenarnya dari data tersebut adalah “anotator” lainnya. Oleh karena itu, *Cohen's Kappa Score* digunakan untuk mengukur seberapa baik model BERT “setuju” dengan label sebenarnya.

Proses evaluasi dalam penelitian ini dilakukan dengan mengumpulkan label sebenarnya (`y_true`) dan label prediksi (`y_pred`) pada data uji, dan kemudian menggunakan fungsi `cohen_kappa_score` dari modul *sklearn.metrics* untuk menghitung nilai *Cohen's Kappa Score*. Metrik evaluasi ini mempertimbangkan tingkat kesepakatan antara label prediksi dan label sebenarnya, sambil memperhitungkan tingkat kesepakatan yang terjadi secara kebetulan. Ini merupakan langkah penting dalam memvalidasi dan mengukur keandalan model

dalam mengklasifikasikan data dengan klasifikasi multilabel. Contoh perhitungan dengan menggunakan *kappa score* :

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

$$K = \frac{0.8 - 0.25}{1 - 0.25}$$

$$K = \frac{0.55}{0.75}$$

$$K = 0.73$$

Dalam perhitungan tersebut untuk mendapatkan  $\Pr(a)$  dengan cara melihat rasio jumlah kesepakatan yang diamati antara dua pengamat terhadap jumlah total observasi dan kemudian untuk  $\Pr(e)$  dihitung berdasarkan probabilitas kedua pengamat setuju secara kebetulan.

### 3.7.3 Accuracy dan Loss Function

Akurasi (accuracy) dan fungsi kerugian (loss function) digunakan sebagai metrik evaluasi yang saling melengkapi dalam proses pelatihan dan evaluasi model machine learning, khususnya pada model BERT untuk klasifikasi teks esai. Akurasi digunakan untuk mengukur seberapa banyak prediksi yang benar dari total prediksi yang dilakukan oleh model, sementara *loss function* mengukur seberapa besar kesalahan antara prediksi model dengan label sebenarnya. Selama pelatihan, model dioptimalkan dengan meminimalkan *loss function*, yang memberikan umpan balik tentang seberapa jauh prediksi menyimpang dari label sebenarnya, sehingga model dapat disesuaikan untuk memperbaiki prediksi di masa depan. Di sisi lain, akurasi digunakan untuk mengevaluasi performa model setelah dilatih, memberikan gambaran tentang seberapa baik model mengklasifikasikan teks esai sesuai dengan label yang diberikan.

## 3.9 Prediksi Model BERT

Prediksi Model BERT ini dilakukan untuk memanfaatkan kekuatan model bahasa *pre-trained* yang kuat, mengkombinasikan metrik yang berbeda untuk penilaian yang lebih komprehensif dan memberikan umpan balik yang bermanfaat bagi penulis esai. Prediksi model BERT yang akan dilakukan untuk menambah penilaian skor yang lebih efisien dengan memanfaatkan:

### 3.8.1 BERT Score

Model BERT yang telah dilatih dapat digunakan dalam berbagai tugas pemrosesan bahasa alami, termasuk klasifikasi teks. Dengan pemanfaatan representasi kontekstual yang kuat dari BERT, akan mendapatkan skor prediksi yang akurat untuk esai berdasarkan konten dan strukturnya. Skor BERT dibutuhkan untuk memanfaatkan kemampuan model BERT dalam memahami dan menganalisis teks secara mendalam. Skor ini memberikan penilaian yang lebih kompleks dan mencakup berbagai aspek seperti semantik dan konteks. Model BERT yang telah dilatih untuk menghasilkan skor prediksi dari esai yang akan diberikan. Inputan esai akan diencode dan diproses oleh model BERT dan skor diambil dari output logit model.

### 3.8.2 Text Similarity

Dalam melakukan prediksi model BERT, kemiripan teks dibutuhkan untuk mengukur seberapa mirip esai yang dievaluasi dengan *prompt* atau esai referensi yang diberikan. Hal ini penting untuk menilai seberapa relevan isi esai dengan topik atau instruksi yang diminta. *Text similarity* ini menggunakan model Word2Vec yang telah dilatih untuk mendapatkan representasi vektor dari setiap kata dalam esai, *prompt*, dan esai referensi. Kemudian, vektor-vektor tersebut digabungkan menjadi satu vektor rata-rata. Lalu, vektor tersebut dihitung kemiripan cosine antara vektor esai dengan vektor prompt dan setiap vektor referensi. Nilai rata-rata dari semua *cosine similarity* tersebut menjadi skor kemiripan teks. Pada penelitian ini akan diberikan beberapa parameter untuk melakukan perhitungan *similarity* antara teks dengan referensi esai yang ada, yaitu :

1. Vektorisasi Teks

Teks esai dan referensi yang ada akan diubah menjadi vektor menggunakan model Word2Vec dengan cara memecah teks menjadi kata-kata, mengonversi setiap kata menjadi vektor dan menghitung rata-rata dari vektor kata-kata untuk mendapatkan satu vektor perwakilan untuk seluruh teks.

2. Penanganan Kata yang Tidak Dikenal

Jika kata tidak ditemukan pada vokabuler Word2Vec, kata tersebut kemudian akan diabaikan. Hal ini membuat hanya kata-kata yang dikenali oleh model yang akan digunakan dalam membentuk vektor teks.

### 3. Cosine Similarity

Setelah teks dikonversi menjadi vektor, kesamaan antara vektor esai dengan vektor prompt dan vektor esai referensi dihitung menggunakan cosine similarity. Cosine similarity mengukur kesamaan dengan menghitung kosinus sudut antara dua vektor. Nilai cosine similarity berkisar dari -1 (sangat tidak mirip) hingga 1 (sangat mirip), dengan nilai yang lebih tinggi menunjukkan tingkat kesamaan yang lebih besar.

### 4. Agregasi Skor

Jika tidak ada kata yang dikenali oleh model Word2Vec dalam esai atau esai referensi, skor default akan dikembalikan. Kemudian, jika ada kata yang dikenali maka rata-rata dari nilai cosine similarity dihitung untuk memberikan skor keseluruhan teks.

#### 3.8.3 Rubrik Skor

Rubrik skor dibutuhkan untuk menilai kualitas esai berdasarkan kriteria tertentu, seperti panjang kalimat rata-rata, koherensi kalimat, dan keberadaan ide utama. Skor ini memberikan penilaian yang lebih objektif dan terstruktur. Rubrik skor akan mengambil nilai dari komponen-komponen yang telah ditetapkan yang kemudian akan dinormalisasi ke rentang 0 hingga 1. Kemudian, setiap komponen dikalikan dengan bobot yang ditentukan untuk setiap komponen dan dijumlahkan untuk mendapatkan hasil akhir yang akan dibatasi ke rentang 0 hingga 100 dalam mendapatkan rubrik skor. Pada penelitian ini akan digunakan beberapa parameter yang perlu untuk mendapatkan nilai rubrik skor, yaitu:

#### 1. Panjang Rata-Rata Kalimat (Average Sentence Length)

Hal ini mengukur panjang rata-rata kalimat dalam esai. Panjang kalimat ini bisa menjadi indikator kemampuan penulis dalam menyusun kalimat yang efektif dan informatif. Bobot yang diberikan untuk panjang kalimat ini senilai 0.4 yang kemudian akan dikalikan dengan hasil dari normalisasi panjang kalimat.

## 2. Koherensi Kalimat (Sentence Coherence)

Koherensi kalimat digunakan untuk mengukur seberapa logis dan terhubung kalimat-kalimat dalam esai tersebut. Ini bisa diukur melalui berbagai metode, seperti melihat transisi antar kalimat atau melalui penilaian manual. Bobot yang diberikan untuk koherensi kalimat ini senilai 0.2 kemudian dilakukan perkalian dengan nilai normalisasi *sentence coherence*.

## 3. Skor Ide Utama (Main Idea Score)

Hal ini mengukur seberapa baik esai menggambarkan atau membahas ide utama yang disajikan dalam *prompt*. Biasanya, ini dihitung dengan menganalisis frekuensi dan relevansi kata kunci yang muncul dalam esai yang juga berhubungan dengan *prompt* atau esai referensi. Bobot yang diberikan untuk koherensi kalimat ini senilai 0.4 kemudian dilakukan perkalian dengan nilai normalisasi *idea score*.

### 3.8.4 Feedback

Penilaian skor atau nilai terhadap suatu esai membutuhkan umpan balik untuk memberikan komentar atau saran kepada penulis esai berdasarkan skor akhir yang diperoleh. Hal ini akan membantu penulis memahami kekuatan dan kelemahan tulisannya. Pada penelitian ini, feedback menerima skor akhir sebagai inputan dan mengembalikan umpan balik dalam bentuk kalimat berdasarkan rentang skor tertentu. Umpan balik ini memberikan penilaian kualitatif tentang kualitas esai dan saran untuk perbaikan.

Setelah semua komponen diatas dihitung, skor akhir akan diperoleh dengan mengkombinasikan skor BERT, kemiripan teks dan skor rubrik. Umpan balik (feedback) juga akan diberikan berdasarkan skor akhir, dengan cara ini penilaian esai menjadi lebih komprehensif dan mempertimbangkan berbagai aspek penting seperti isi, struktur, dan kualitas tulisan.

Feedback akan diberikan sesuai dengan dua pendekatan, yaitu feedback holistik dan feedback analitik. Feedback holistik akan didasarkan pada penilaian skor akhir, memberikan gambaran umum tentang kualitas keseluruhan esai. Sementara itu, feedback analitik akan disesuaikan dengan nilai-nilai dari parameter rubrik skor yaitu panjang kalimat, koherensi kalimat, ide utama dan nilai hasil kemiripan teks, yang akan memberikan umpan balik lebih terperinci

untuk setiap aspek penilaian. Dengan pendekatan ini, penilaian esai akan menjadi lebih holistik dan memperhitungkan aspek-aspek yang beragam dalam penulisan.

### **3.10 Hasil Keluaran Sistem**

Setelah melewati serangkaian proses, penelitian ini menghasilkan sebuah skor akhir yang merupakan gabungan dari tiga komponen utama, yaitu skor BERT, kemiripan teks, dan skor rubrik. Skor akhir tersebut dihitung dengan mempertimbangkan bobot khusus untuk setiap komponen, sehingga memberikan penilaian yang lebih holistik terhadap kualitas esai yang dianalisis. Selain skor akhir, penelitian juga menghasilkan umpan balik berupa kalimat atau komentar yang bersifat kualitatif, memberikan evaluasi terperinci mengenai kekuatan dan kelemahan tulisan, serta area yang perlu diperbaiki. Dengan demikian, hasil akhir yang diperoleh dari penelitian ini tidak hanya berupa angka representatif dari skor akhir esai, tetapi juga disertai dengan umpan balik yang berharga bagi penulis esai dalam meningkatkan kualitas tulisannya.

Penelitian ini menggunakan berbagai metrik dan model, termasuk BERT untuk pemahaman bahasa, Word2Vec untuk analisis kemiripan semantik, dan kriteria rubrik untuk menilai aspek-aspek seperti struktur kalimat, koherensi, dan kejelasan ide utama. Dengan pendekatan ini, diharapkan hasil evaluasi dapat memberikan pandangan yang holistik dan menyeluruh terhadap kualitas esai, serta memberikan bimbingan yang berguna bagi penulis dalam upaya meningkatkan kemampuan menulisnya.

### **3.11 Perancangan Sistem**

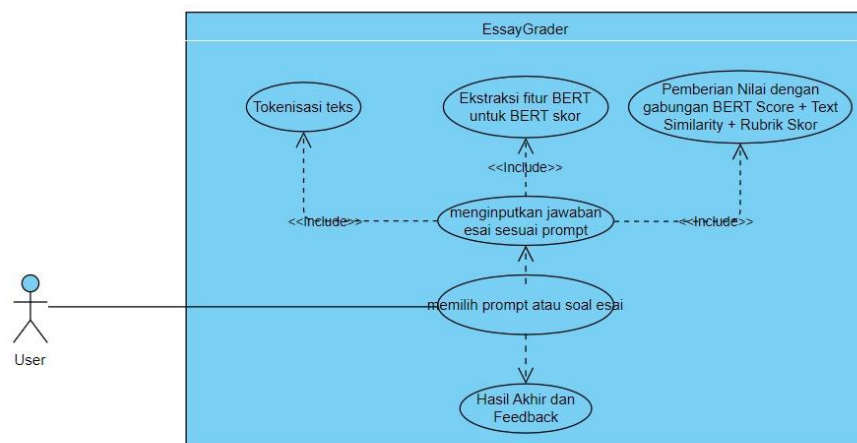
Dalam merancang sistem yang diinginkan, diperlukan beberapa diagram yang menggambarkan secara detail spesifikasi sistem yang direncanakan. Selain itu, perlu dipertimbangkan alur proses dari tahap perancangan program untuk memastikan konsistensi dan keselarasan yang diperlukan.

#### **3.10.1 Use Case Diagram**

*Use Case Diagram* adalah jenis diagram di *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna eksternal atau sistem lain) dan sistem yang sedang dirancang. Diagram ini membantu dalam memahami fungsionalitas sistem dengan menunjukkan



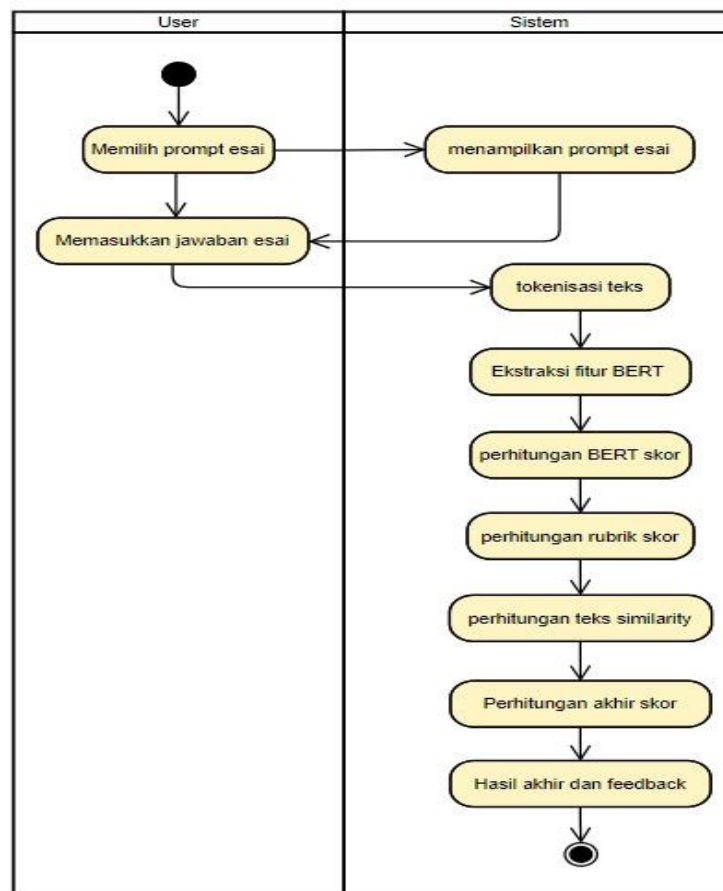
tindakan atau kasus penggunaan yang berbeda serta hubungan antara aktor dan kasus penggunaan. *Use case diagram* memberikan pandangan yang lebih luas tentang sistem dengan menyoroti tujuan utama pengguna dan bagaimana sistem akan digunakan dalam skenario berbeda. Interaksi yang dilakukan oleh *user* pada sistem dapat dilihat pada gambar 3.5.



**Gambar 3. 8** *Use Case Diagram*

### 3.10.2 Activity Diagram

Diagram aktivitas merupakan jenis diagram yang terdapat dalam *Unified Modeling Language* (UML) yang berfungsi untuk mengilustrasikan alur kerja atau proses bisnis dari suatu sistem atau bagian dari sistem. Diagram ini menampilkan serangkaian aktivitas yang terjadi dalam suatu proses, serta menunjukkan urutan dan kondisi yang mengarah pada transisi antara aktivitas-aktivitas tersebut. Dengan menggunakan *activity diagram*, pemangku kepentingan dapat memodelkan logika bisnis dan alur kerja sistem secara visual, sehingga memudahkan pemahaman tentang bagaimana suatu proses bekerja dan membantu dalam mengidentifikasi area untuk perbaikan atau peningkatan yang diperlukan. *Activity diagram* penelitian ini dapat dilihat pada gambar 3.6.

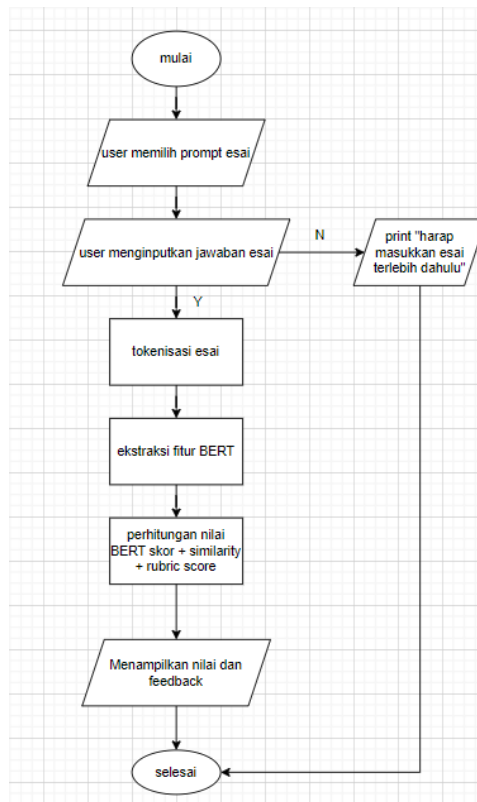


**Gambar 3.9** Activity Diagram

Berdasarkan gambar 3.6 di atas, *activity diagram* dimulai dari user dapat memilih prompt esai yang ingin di isi terlebih dahulu, lalu sistem akan menampilkan *prompt* esai tersebut dan setelah itu user dapat memasukkan jawaban esai yang sesuai dan esai tersebut dilakukan tokenisasi teks. Kemudian dilakukan ekstraksi fitur BERT untuk melakukan perhitungan BERT skor yang kemudian akan dilakukan kembali perhitungan *text similarity* antara teks referensi esai dengan jawaban esai yang diinputkan user. Lalu setelah itu dilakukan perhitungan rubrik skor dengan menghitung panjang kalimat, koherensi kalimat dan ide utama pada jawaban esai. Setelah itu sistem akan melakukan perhitung akhir skor yang meliputi BERT skor, *similarity* skor dan rubrik skor sehingga sistem akan menampilkan hasil akhir dan *feedback* terhadap jawaban esai user.

### 3.10.3 Flowchart Sistem

*Flowchart* adalah representasi visual dari alur logika sebuah proses atau sistem, menggunakan simbol-simbol standar untuk menampilkan langkah-langkah proses, keputusan, dan aliran informasi dari awal hingga akhir. *Flowchart* juga bermanfaat untuk mendokumentasikan, menganalisis, dan memperbaiki proses-proses yang ada, serta merencanakan proses-proses baru.



**Gambar 3. 10** *Flowchart Sistem*

Pada gambar 3.7 di atas menjelaskan *flowchart* dari sistem yang dimulai dari user memilih *prompt* esai kemudian user menginputkan jawaban esai yang akan diproses dengan dilakukan tokenisasi jawaban esai. Lalu fitur dari kalimat esai akan di ekstraksi menggunakan BERT dan dilakukan perhitungan nilai dengan BERT skor, *similarity* dan rubrik skor. Kemudian *website* akan menampilkan nilai dan *feedback* terhadap jawaban esai yang diberikan.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi Sistem

Setelah tahap analisis dan perancangan, tahap selanjutnya adalah implementasi dan pengujian, spesifikasi perangkat keras dan lunak yang digunakan pada penelitian ini sebagai berikut :

1. *Processor* 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz.
2. RAM 16GB.
3. Sistem operasi *Windows 11 Home Single Language* 64-bit.
4. *Browser Google Chrome* dan *Opera Browser*.
5. *Google Colaboratory Pro*
6. *Python 3.7*.
7. *Library* pandas, numpy, nltk, sklearn, gensim, transformers, tensorflow, torch, matplotlib, scikit-learn dan streamlit.

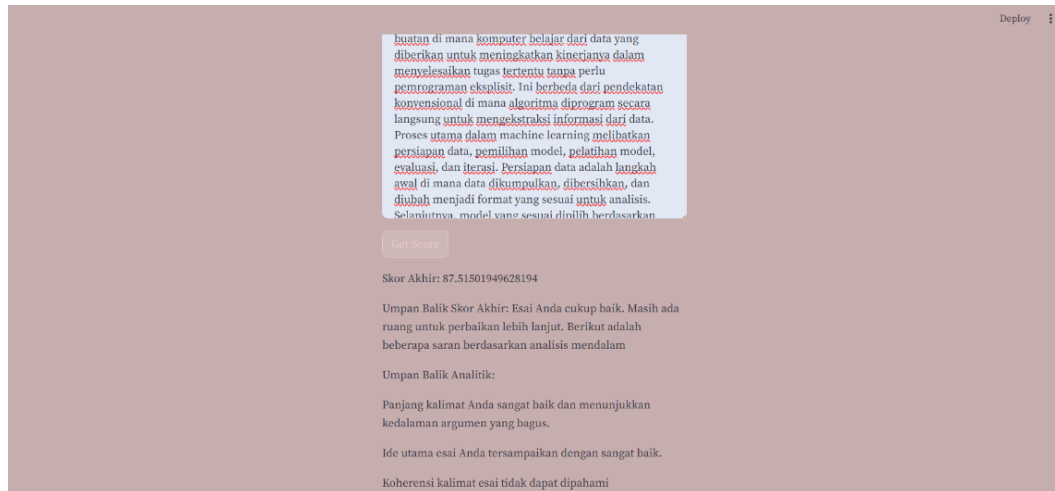
#### 4.2 Implementasi Website

Berikut ini merupakan hasil implementasi sistem dalam website. Hal ini dimulai dengan menampilkan halaman awal dari website sistem yang dibuat.



**Gambar 4. 1** Halaman Utama

Pada halaman ini user memilih *prompt* atau soal esai yang ingin dijawab dan kemudian akan dialihkan ke halaman yang dimana user dapat menginputkan jawaban esai sesuai dengan *prompt* atau soal yang dipilih.



**Gambar 4. 2** Hasil Prediksi Esai

Setelah user memilih soal yang akan dijawab, user akan mengetikkan jawaban esai yang sesuai dengan yang diminta oleh dosen atau *prompt* yang tertera. Lalu, setelah itu sistem akan melakukan proses pemeriksaan esai untuk mendapatkan nilai dan umpan balik terhadap jawaban esai yang telah di input. Hasil yang akan dikeluarkan oleh sistem berupa nilai dan umpan balik.

### 4.3 *Pre-Processing* dataset

Tahap *pre-processing* pada sistem ini terdiri dari beberapa proses yaitu, *stemming*, *stop word removal*, dan *special character removal* dataset. Proses *stemming* dan *stop word removal* pada sistem ini dilakukan dengan menggunakan *library* nltk. Tahapan yang dilakukan pada *pre-processing* sistem ini dilakukan inisialisasi terhadap proses-proses yang akan dilakukan agar dapat dilakukan tahapan untuk memproses seluruhnya agar mendapatkan hasil dataset yang sudah bersih.

```
[ ] def preprocess_text(text):
    text = remove_special_chars(text)
    stemmed_text = stem_text(text)
    filtered_text = remove_stop_words(stemmed_text)
    return filtered_text
```

**Gambar 4. 3** *Preprocess text*

Gambar 4.3 merupakan insialisasi seluruh proses yang akan dilakukan dalam melakukan *pre-processing* dataset. Proses ini bertujuan untuk membersihkan teks dari *noise*, mengurangi variasi kata yang tidak relevan dan mempersiapkan teks untuk analisis lebih lanjut. Implementasi *pre-processing* ini dapat dilihat dari gambar 4.4 sebagai hasil dari implementasi pembersihan dataset dengan *stemming*, *stop word removal* dan *special character removal*.

| 1  |    | essay  | domain1 | score |
|----|----|--|---------|-------|
| 2  | 0  | surat kabar lokal terhormat pengaruh komput manusia keterampilanpengaruh belajar hebat kompu |         |       |
| 3  | 1  | dear caps1 caps2 percaya komput manfaat berbicara berteman orang situ web facebook mysac kom |         |       |
| 4  | 2  | terhormat caps1 caps2 caps3 orang komput orang setuju bermanfaat masyarakat mendukung kemaju |         |       |
| 5  | 3  | koran lokal terhormat caps1 menemukan ahli komput manfaat masyarakat kasu kasu penelitian ko |         |       |
| 6  | 4  | location1 terhormat memiliki komput dampak positif manusia komput menghubungkan keluarga ber |         |       |
| 7  | 5  | terhormat location1 komput dampak negatif orang memiliki aks kampu amerika num1 orang mengak |         |       |
| 8  | 6  | tahukah orang bergantung komput keselamatan pendidikan alam kehidupan sosial meningkatnya pe |         |       |
| 9  | 7  | percent1 orang setuju komput hidup mudah setuju penggunaan komput mengajarkan koordinasi tar |         |       |
| 10 | 8  | pembaca budiman organization1 dampak dramati kehidupan manusia mengubah pale komput perangka |         |       |
| 11 | 9  | location1 memiliki teknolog komput komput bermanfaat masyarakat setuju buruk alasan penyebab |         |       |
| 12 | 10 | location1 terhormat orangorang caps1 mengakui kemajuan komput menempatkan kelemahan caps1 ko |         |       |
| 13 | 11 | dear caps1 caps2 komput merampa kehidupan manusia sepent faktor kehidupan pertamamata dunia  |         |       |
| 14 | 12 | surat kabar lokal terhormat membaca argumen komput argumen berdampak positif manusia alasan  |         |       |

**Gambar 4. 4** Hasil *Pre-processing* dataset

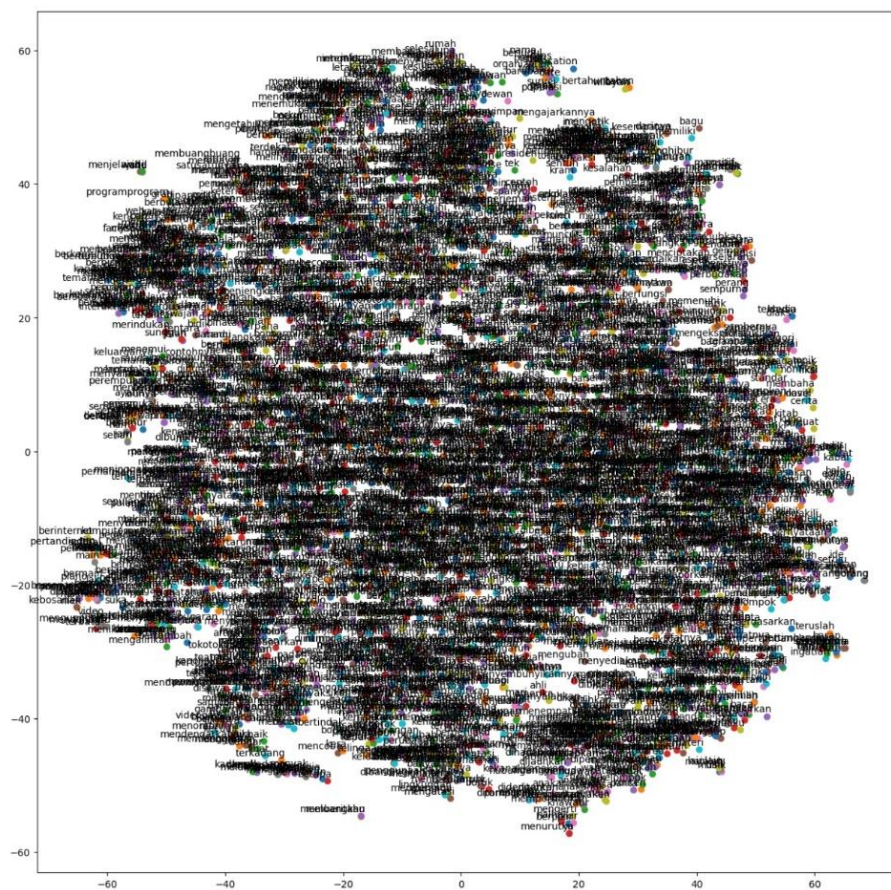
#### 4.4 Representasi Vektor Word2Vec

Setelah melakukan tahap *pre-processing* dataset, dilakukan juga tahap untuk mendapatkan representasi vektor Word2vec dari setiap kalimat yang ada pada dataset. Dilakukan pelatihan model Word2Vec dengan menggunakan data kalimat pada dataset dengan memanfaatkan parameter-parameter yang digunakan. Kemudian pada gambar 4.4 proses yang dilakukan bertujuan untuk menghasilkan vektor fitur dari kata-kata dalam teks menggunakan model word2vec. Fungsi *'makeFeatureVec()'* menghasilkan vektor fitur dari setiap kata dalam teks. Fungsi *'getAvgFeatureVecs()'* mengaplikasikan proses ini pada Kumpulan teks dan menghasilkan vektor fitur untuk setiap teks. Hasilnya adalah matriks yang berisi vektor fitur untuk setiap teks dalam kumpulan esai yang diberikan.

```
[ ] def getAvgFeatureVecs(essays, model, num_features):
    """Main function to generate the word vectors for word2vec model."""
    counter = 0
    essayFeatureVecs = np.zeros((len(essays), num_features), dtype="float32")
    for essay in essays:
        essayFeatureVecs[counter] = makeFeatureVec(essay, model)
        counter += 1
    return essayFeatureVecs

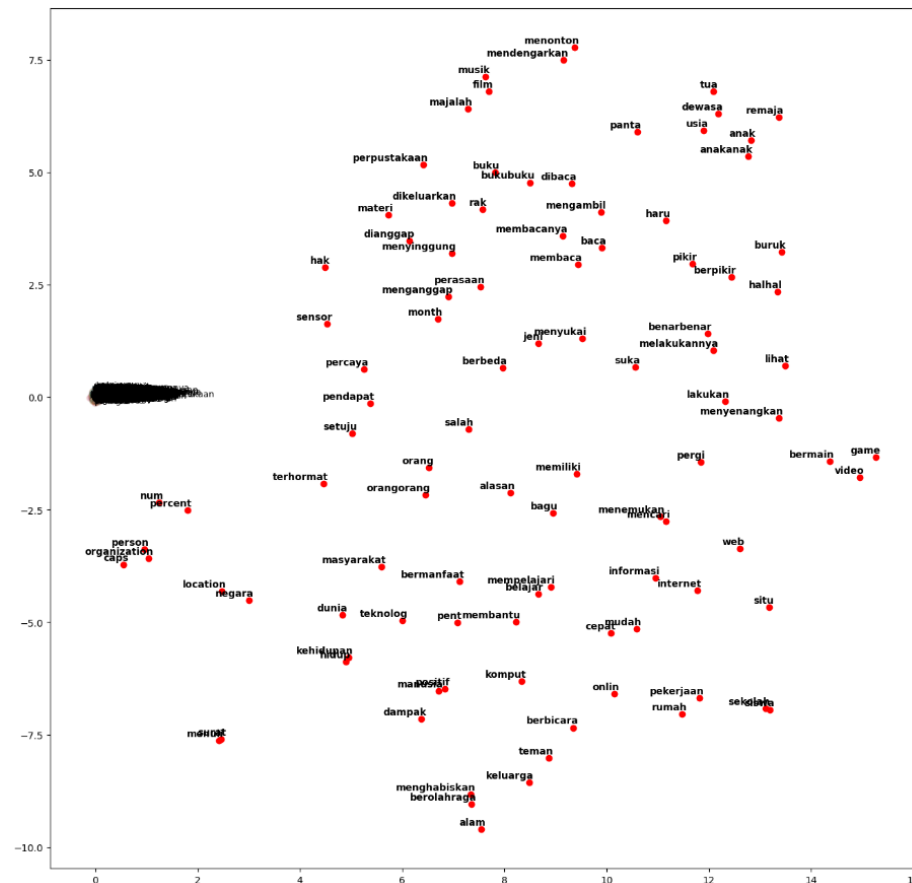
[ ] def makeFeatureVec(words, model):
    """Make Feature Vector from the words list of an Essay."""
    num_features = model.vector_size
    featureVec = np.zeros((num_features,), dtype="float32")
    num_words = 0
    for word in words:
        if word in model.wv.key_to_index: # Periksa apakah kata ada dalam kamus model
            num_words += 1
            featureVec = np.add(featureVec, model.wv.get_vector(word)) # Menggunakan get_vector() untuk mendapatkan vektor kata
    if num_words > 0:
        featureVec = np.divide(featureVec, num_words)
    return featureVec
```

Gambar 4. 5 Representasi Vektor



Gambar 4. 6 Visualisasi model Word2Vec





**Gambar 4. 7** Top 100 Vektor Word2Vec

Setelah melakukan proses untuk mendapatkan vektor fitur dari kata-kata dari dataset, maka dilakukan proses *training model* hingga *save model* yang berisi representasi vektor-vektor dari dataset dengan memanfaatkan model Word2Vec. Oleh karena itu, pada sistem ini akan menampilkan visualisasi hasil dari model Word2Vec yang telah dibangun. Dalam visualisasi tersebut, kata-kata yang memiliki makna atau konteks yang mirip akan cenderung berada dekat satu sama lain, sedangkan kata-kata yang berbeda makna atau konteks akan cenderung berada jauh satu sama lain. Hal ini memungkinkan untuk melihat pola hubungan antara kata-kata dalam ruang fitur, yang dapat memberikan wawasan yang berguna tentang struktur semantik dalam korpus teks yang digunakan untuk melatih model word2vec.

#### 4.5 Implementasi Algoritma BERT

Setelah melakukan tahapan representasi vektor dengan model Word2Vec, dilakukan implementasi Algoritma BERT *fine-tuning*. *Fine-tuning* merupakan



proses pengambilan model BERT yang telah dilatih sebelumnya (*pre-trained*) dan melakukan pelatihan lebih lanjut untuk tugas penilaian esai otomatis.

```
[ ] from transformers import BertTokenizer, BertModel

tokenizer = ppb.BertTokenizer.from_pretrained(bert_model_path)
model = ppb.BertForSequenceClassification.from_pretrained(bert_model_path, num_labels=1)
model.to(cuda)
```

**Gambar 4. 8** Pemuatan Model BERT

Pada gambar 4.8 diatas, program menggunakan *library* Transformers untuk menginisialisasi sebuah tokenizer BERT dan sebuah model BERT untuk klasifikasi teks. Pada kodingan memuat tokenizer dan model BERT yang sudah dilatih sebelumnya '*from\_pretrained*' dari jalur '*bert\_model\_path*' yaitu model *bert-base-uncased*. Model yang dimuat adalah *BertForSequenceClassification*, yang merupakan model BERT yang dikhususkan untuk tugas klasifikasi sekuens. Dengan memuat model yang sudah dilatih sebelumnya, sistem memanfaatkan pengetahuan yang telah diperoleh model BERT sebelumnya.

```
def collate_fn(batch, tokenizer=tokenizer):
    input_ids, attention_masks, labels = zip(*batch)
    input_ids = pad_sequence([ids.clone().detach() for ids in input_ids], batch_first=True, padding_value=tokenizer.pad_token_id)
    attention_masks = pad_sequence([mask.clone().detach() for mask in attention_masks], batch_first=True, padding_value=0)
    labels = torch.tensor(labels, dtype=torch.float)
    return input_ids, attention_masks, labels
```

**Gambar 4. 9** *Collate\_fn*

Kodingan untuk fungsi '*collate\_fn*' ini digunakan untuk memformat *batch* data input sebelum diberikan kepada model BERT dalam proses pelatihan. Fungsi ini melakukan pengelompokkan *input\_ids* dan *attention\_masks* dalam batch ke dalam tensor dengan *padding*, serta mengonversi label menjadi tensor dengan tipe data *float*. Hal ini memastikan bahwa setiap *batch* memiliki format yang konsisten dan siap untuk diproses oleh model BERT dalam pelatihan. Dataset esai dan label dalam format yang sesuai untuk pelatihan atau evaluasi model.

```

# Train the model
model.train()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-5)

fold_train_losses = []
fold_train_accuracies = []

for epoch in range(epochs):
    running_loss = 0.0
    correct_predictions = 0
    total_predictions = 0

```

**Gambar 4. 10** *Training Dataset dengan Fine-tuning BERT*

Pada bagian ini, dilakukan pelatihan (*fine-tuning*) pada model BERT yang telah dimuat sebelumnya. Tahapan ini dilakukan pengaturan *optimizer*, melakukan *loop* pelatihan, dan menghitung *loss* serta akurasi selama proses pelatihan. Dengan melakukan pelatihan pada data esai dan label skor yang spesifik, model BERT akan menyesuaikan bobot dan parameter internalnya untuk tugas klasifikasi esai tersebut. Dalam proses grading teks esai menggunakan BERT, input yang diperlukan adalah teks esai itu sendiri. Teks esai tersebut akan diproses oleh model BERT dengan melakukan tokenisasi, yaitu memecah teks menjadi token-token yang dapat dipahami oleh model. BERT kemudian akan menghasilkan representasi vektor untuk setiap token berdasarkan konteks di dalam teks esai. Representasi vektor ini digunakan untuk memprediksi skor esai yang sesuai dengan teks yang diberikan sebagai output. Output dari BERT berupa skor numerik yang merepresentasikan penilaian terhadap teks esai tersebut. Selama proses pelatihan, prediksi skor dari model BERT akan dibandingkan dengan skor sebenarnya yang sudah diketahui sebelumnya untuk menghitung *loss* dan memperbarui bobot model agar akurasi prediksi semakin meningkat.

#### 4.6 Hasil Hyperparameter Tuning

Penelitian ini melalui proses *hyperparameter tuning* yang telah dijelaskan skenarionya pada tabel 3.4. Proses ini dilakukan dengan melatih model untuk seluruh dataset. Hasil dari setiap *hyperparameter tuning* dengan *k-fold cross-validation* disajikan oleh tabel 4.1.

**Tabel 4. 1** Hasil *Hyperparameter Tuning*

| Epoch | Batch Size | Learning Rate | K-Fold Cross Validation | Avg Accuracy | Avg Kappa Score |
|-------|------------|---------------|-------------------------|--------------|-----------------|
| 20    | 8          | 2e-5          | 3                       | 0.94         | 0.68            |
| 30    | 8          | 3e-5          | 3                       | 0.85         | 0.65            |
| 30    | 8          | 1e-5          | 4                       | 1            | 0.82            |

Dalam *hyperparameter* ini yang perlu dilihat merupakan hasil dari rata-rata *kappa score* yang lebih tinggi dan juga *accuracy* yang sempurna sehingga dapat menghasilkan model yang baik yang akan digunakan dalam sistem ini. Oleh karena itu, dalam mencapai model yang baik *hyperparameter* yang digunakan disajikan pada tabel 4.2 ini.

**Tabel 4. 2** *Hyperparameter tuning model*

| Hyperparameter          | Value  |
|-------------------------|--------|
| Epoch                   | 30     |
| Batch size              | 8      |
| Optimizer               | Adam   |
| Learning Rate           | 1e-5   |
| K-Fold Cross Validation | 4 Fold |

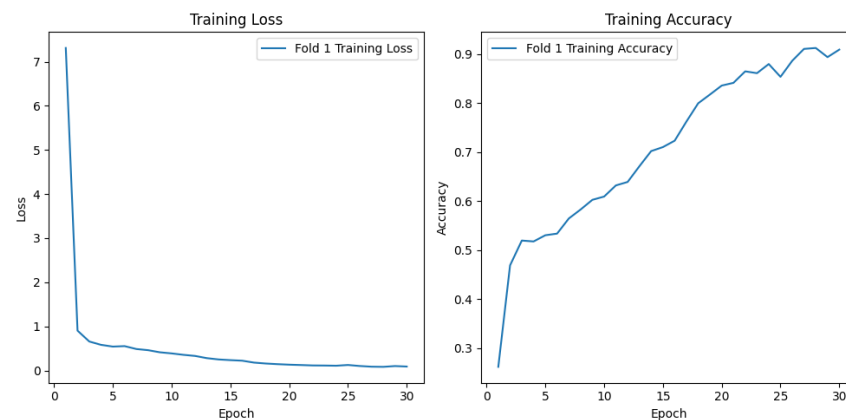
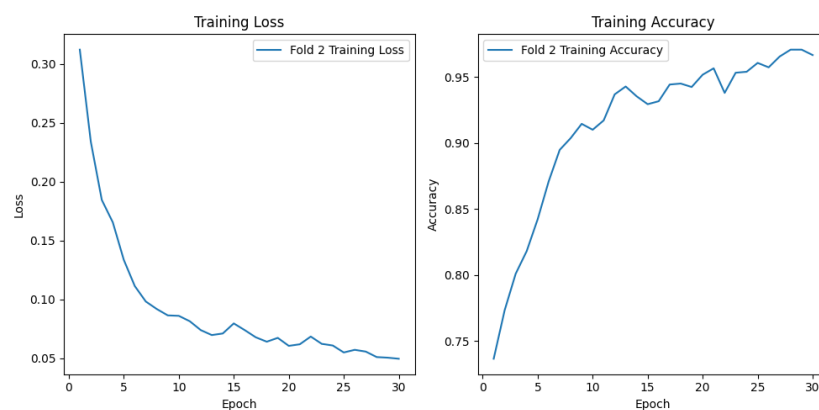
#### 4.7 Evaluasi

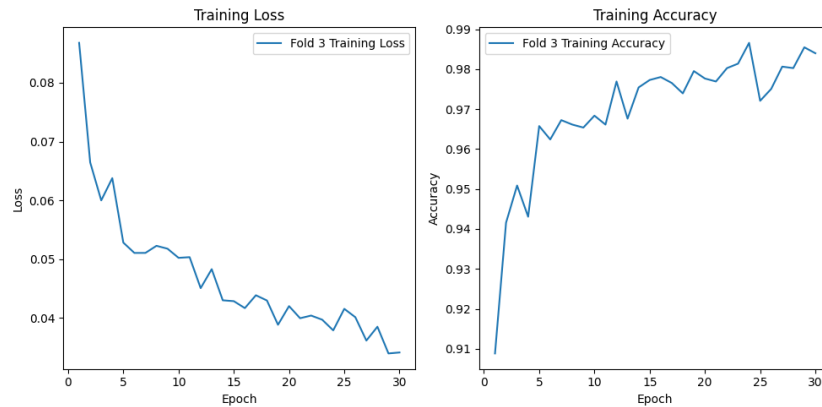
Dalam penelitian ini, diperlukan adanya proses evaluasi terhadap model yang telah dibuat. Setelah proses implementasi algoritma BERT, kemudian dilakukan proses evaluasi terhadap kinerja model BERT untuk tugas klasifikasi esai yang akan dilakukan dengan menggunakan metrik *Kappa Score* dan Akurasi. Proses evaluasi ini juga menggunakan pendekatan *K-Fold Cross-Validation* untuk mendapatkan estimasi kinerja model. Model dievaluasi menggunakan data validasi dengan memanfaatkan ‘y\_pred’ dan y>true’ dalam menghitung akurasi dengan membandingkan prediksi dengan label sebenarnya sehingga menghasilkan *accuracy* setiap *fold* dan rata-ratanya seperti pada tabel 4.3 berikut

**Tabel 4. 3 Accuracy**

| K-Fold Cross Validation | Accuracy |
|-------------------------|----------|
| 1                       | 0.5290   |
| 2                       | 0.8795   |
| 3                       | 0.9877   |
| 4                       | 0.9978   |
| <b>Average</b>          | <b>1</b> |

Dapat dilihat bahwa setiap *fold* yang dievaluasi terdapat peningkatan *accuracy* yang stabil, sehingga dapat diketahui bahwa algoritma BERT memiliki tingkat akurasi yang tinggi yaitu 1 pada seluruh dataset. Untuk visualisasi *accuracy* dan *loss function* setiap *fold* dapat dilihat pada gambar 4.11, 4.12, 4.13 dan 4.14 ini.

**Gambar 4. 11 Accuracy fold pertama****Gambar 4. 12 Accuracy fold kedua**



**Gambar 4. 13** *Accuracy fold ketiga*



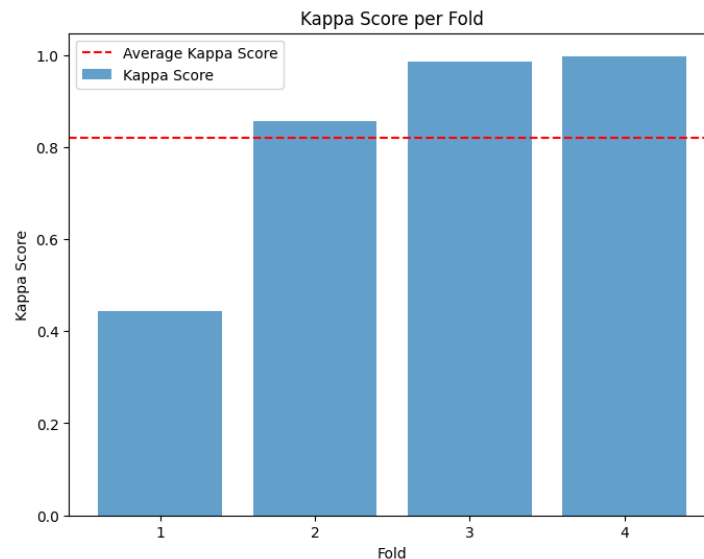
**Gambar 4. 14** *Accuracy fold keempat*

Kemudian, dalam melakukan evaluasi terhadap model BERT tidak hanya dilakukan dengan *accuracy* saja. BERT untuk penilaian esai otomatis memerlukan *kappa score* dalam menentukan model BERT sudah bisa mempelajari seluruh dataset yang telah dikumpulkan sebelumnya. Hasil dari *kappa score* untuk dataset ini dapat dilihat pada tabel 4.4.

**Tabel 4. 4** *Kappa Score*

| <i>K-Fold Cross Validation</i> | <i>Kappa Score</i> |
|--------------------------------|--------------------|
| 1                              | 0.4428             |
| 2                              | 0.8568             |
| 3                              | 0.9855             |
| 4                              | 0.9973             |
| <i>Average</i>                 | 0.82               |

Hasil *kappa score* tersebut diperoleh dengan melakukan evaluasi sebanyak empat *fold* dan kemudian dilakukan rata-rata untuk seluruh *fold*. Sehingga menghasilkan nilai rata-rata *kappa score* nya yaitu 0.82 untuk seluruh dataset.



**Gambar 4. 15** Grafik *Kappa Score*

Grafik ini memberikan gambaran tentang variasi kinerja model dalam hal *Kappa Score* pada setiap *fold* dalam proses evaluasi *k-fold cross-validation*.

## 4.8 Prediksi Model BERT

Hasil dari evaluasi model BERT kemudian dilakukan *save model* untuk bisa digunakan dalam melakukan prediksi esai dengan model BERT yang sudah dilatih. Pada tahapan prediksi model BERT ini memanfaatkan beberapa hal untuk mendapatkan hasil yang baik untuk skor esai.

### 4.6.1 BERT Score

Pada tahapan ini dilakukan untuk mengukur kesamaan semantik antara teks esai dengan referensi atau model yang telah dilatih sebelumnya menggunakan model BERT.

```

def get_bert_score(essay, model, tokenizer, cuda):
    encoding = tokenizer(essay, truncation=True, padding=True, max_length=512, return_tensors='pt')
    input_ids, attention_mask = encoding['input_ids'].to(cuda), encoding['attention_mask'].to(cuda)

    with torch.no_grad():
        outputs = model(input_ids, attention_mask=attention_mask)
        score = outputs.logits.cpu().numpy().flatten()[0]

    # Normalisasi skor ke rentang 0 hingga 100
    score = (score + 10) / 20 * 100 # Asumsi skor minimum adalah -10 dan maksimum 10
    score = max(score, 0) # Memastikan skor tidak negatif
    score = min(score, 100) # Memastikan skor tidak melebihi 100

    return int(round(score))

```

**Gambar 4. 16** Prediksi BERT Score

Skor untuk BERT score dihitung menggunakan fungsi *get\_bert\_score* yang mengambil input berupa esai, model BERT yang telah dilatih, dan *tokenizer*. Skor BERT mencerminkan seberapa baik esai sesuai dengan konteks dan makna yang diharapkan.

**BERT Score: 70**

**Gambar 4. 17** Hasil BERT Score

Hasil yang dikeluarkan oleh fungsi *get\_bert\_score* tersebut dihitung dengan model BERT yang telah dilatih sehingga mengeluarkan output seperti pada gambar 4.25. Kemudian, hasil tersebut tidak langsung dikeluarkan untuk memberikan penilaian terhadap sebuah esai. Hasil tersebut akan dijumlahkan kembali dengan komponen atau parameter lainnya untuk mendapatkan nilai yang sesuai dengan yang diharapkan.

#### 4.6.2 Text Similarity

Metrik ini mengukur kesamaan antara teks esai dengan prompt atau instruksi yang diberikan, serta dengan esai referensi yang telah disediakan. Pada *similarity* ini, dilakukan pemberian bobot untuk setiap komponen yang digunakan. Dalam memeriksa bagaimana kesamaan antar teks tidak hanya dilihat menggunakan cosine similarity tetapi dengan melihat parameter lain seperti representasi vektor antar esai dan referensi, mencari vektor yang sama pada Word2Vec dan kemudian menggunakan *cosine similarity* untuk melihat seberapa mirip antara jawaban esai dengan referensi esai yang telah diberikan. Skor text similarity dihitung menggunakan fungsi *get\_text\_similarity* yang mengambil input berupa esai,

*prompt*, esai referensi, dan model word2vec (word-to-vector) yang telah dilatih sebelumnya.

```
-4.59618457e-02 -4.15873155e-02 2.60457844e-02 -8.72327853e-03
-2.09457055e-02 1.25508755e-04 -3.95462438e-02 -8.76261108e-03
1.32355113e-02 4.11170907e-02 1.88437141e-02 5.20267934e-02
2.35951841e-02 -3.37547585e-02 -8.03079084e-02 -2.79005896e-02
7.60869831e-02 -2.12605223e-02 2.69178916e-02 1.04376227e-01
-4.42122519e-02 -3.98104675e-02 -9.21988040e-02 6.59639155e-03
-5.89698404e-02 3.74335758e-02 2.68620271e-02 6.41694963e-02
7.35708997e-02 3.31953056e-02 4.31277119e-02 1.15665510e-01
-2.14288682e-02 -3.49770561e-02 4.61934283e-02 7.68324500e-03]
Cosine similarity dengan essay referensi ke-[0]: 0.9277454614639282
Cosine similarity dengan essay referensi ke-[1]: 0.9999998807907104
Skor text similarity: 96.38726711273193
```

**Gambar 4. 18** Hasil *Similarity*

Skor ini mencerminkan seberapa relevan esai dengan topik atau *prompt* yang diberikan. Hal ini dihitung dengan memberikan rata-rata nilai dari representasi vektor, *similarity* dengan esai referensi dan kata pada model Word2Vec dan dilakukan perhitungan rata-rata yang sesuai dengan seluruh komponen.

#### 4.6.3 Rubrik Skor

Rubrik skor ini dihitung berdasarkan beberapa komponen rubrik yang meliputi, *Avg sentence length*, *sentence coherence*, dan *main idea score*.

```
[ ] def calculate_avg_sentence_length(essay):
    # Tokenisasi kalimat
    sentences = nltk.sent_tokenize(essay)

    # Menghitung panjang setiap kalimat
    sentence_lengths = [len(re.findall(r'\w+', sentence)) for sentence in sentences]

    # Menghitung rata-rata panjang kalimat
    avg_sentence_length = sum(sentence_lengths) / len(sentence_lengths)

    return avg_sentence_length
```

**Gambar 4. 19** *Avg Sentence Length*

Fungsi '*calculate\_avg\_sentence\_length*' digunakan untuk menghitung rata-rata panjang kalimat dalam sebuah esai. Hasilnya merupakan rata-rata panjang kalimat dalam esai tersebut yang akan dinilai bersama kriteria lainnya untuk rubrik skor.



```
def calculate_sentence_coherence(essay, prompt, reference_essays):
    # Tokenisasi kalimat dari esai
    essay_sentences = nltk.sent_tokenize(essay)

    # Tokenisasi kalimat dari prompt dan referensi esai
    all_sentences = [nltk.sent_tokenize(str(prompt))] # Konversi prompt ke string
    for ref_essay in reference_essays:
        all_sentences.extend(nltk.sent_tokenize(str(ref_essay))) # Konversi ref_essay ke string

    # Menghitung koherensi antar kalimat esai dengan prompt dan referensi
    coherence_scores = []
    for essay_sent in essay_sentences:
        words_essay = re.findall(r'\w+', str(essay_sent)) # Konversi essay_sent ke string
        max_overlap = 0
        for other_sent in all_sentences:
            words_other = re.findall(r'\w+', str(other_sent)) # Konversi other_sent ke string
            overlap = len(set(words_essay).intersection(set(words_other)))
            max_overlap = max(max_overlap, overlap)
            coherence_score = max_overlap / len(words_essay)
            coherence_scores.append(coherence_score)

    # Menghitung rata-rata koherensi kalimat
    avg_coherence = sum(coherence_scores) / len(coherence_scores) if coherence_scores else 0

    return avg_coherence * 100
```

**Gambar 4. 20** *Calculate Sentence Coherence*

Fungsi `calculate_sentence_coherence` digunakan untuk mengukur tingkat koherensi antara kalimat-kalimat dalam sebuah esai dengan *prompt* dan esai referensi. Hal ini dilakukan dengan menghitung seberapa banyak kata-kata dalam setiap kalimat esai yang tumpang tindih dengan kata-kata dalam kalimat-kalimat *prompt* dan referensi.

```
def check_main_idea(essay, prompt, reference_essays):
    # Mengambil kata kunci dari prompt
    prompt_keywords = re.findall(r'\w+', prompt.lower())

    # Mengambil kata kunci dari referensi esai
    reference_keywords = []
    for ref_essay in reference_essays:
        reference_keywords.extend(re.findall(r'\w+', ref_essay.lower()))

    # Menggabungkan kata kunci dari prompt dan referensi esai
    all_keywords = prompt_keywords + reference_keywords

    # Menghitung frekuensi kemunculan kata kunci dalam esai
    essay_words = re.findall(r'\w+', essay.lower())
    keyword_counts = Counter(essay_words)

    # Menghitung skor main idea berdasarkan kemunculan kata kunci
    total_keyword_count = sum(keyword_counts[keyword] for keyword in all_keywords)
    main_idea_score = float(total_keyword_count) / len(all_keywords)

    return main_idea_score
```

**Gambar 4. 21** *Check Main Idea*

Fungsi `check_main_idea` digunakan untuk mengevaluasi sejauh mana sebuah esai menangkap gagasan utama yang terkandung dalam *prompt* dan esai referensi. Ini

dilakukan dengan menghitung seberapa sering kata kunci dari prompt dan esai referensi muncul dalam esai yang dievaluasi.

```
def compute_rubric_score(avg_sentence_length, sentence_coherence, main_idea_score):
    # Normalisasi setiap komponen ke rentang 0 hingga 1
    normalized_avg_sentence_length = min(avg_sentence_length / 20, 1)
    normalized_sentence_coherence = min(sentence_coherence / 100, 1)
    normalized_main_idea_score = min(main_idea_score, 1)

    # Menghitung skor akhir dengan bobot yang sesuai
    score = (normalized_avg_sentence_length * 0.4) + (normalized_sentence_coherence * 0.2) + (normalized_main_idea_score * 0.4)

    # Membatasi skor ke rentang 0 hingga 100
    rubric_score = int(max(min(score, 1), 0) * 100)

    return rubric_score
```

**Gambar 4. 22** *Compute Rubric Score*

Fungsi `compute_rubric_score` digunakan untuk menghitung skor akhir dari sebuah esai berdasarkan tiga komponen penilaian. Setiap komponen dinormalisasi agar nilainya berada dalam rentang 0 hingga 1. Kemudian, skor akhir dihitung dengan menerapkan bobot yang sesuai untuk setiap komponen. Skor akhir kemudian dibatasi agar nilainya berada dalam rentang 0 hingga 100.

```
Rubric Score: 98
Combined Score (bert_score + text_similarity + rubric_score): 277.59181666374207
Komponen Rubrik:
Average Sentence Length: 1.0
Sentence Coherence: 0.9428900709219857
Main Idea Score: 1
```

**Gambar 4. 23** *Hasil Rubric Score*

Bentuk hasil dari setiap perhitungan parameter rubrik skor yang kemudian seluruh parameter tersebut dilakukan perkalian dengan bobot yang telah diberi sehingga nilai dari komponen tersebut dilakukan rata-rata penilaian untuk mendapatkan hasil yang sesuai.

#### 4.6.4 Feedback

Umpan balik dalam berupa komentar atau saran yang diberikan kepada penulis esai diberikan berdasarkan skor akhir yang diperoleh. Umpan balik ini dihasilkan menggunakan fungsi `generate_feedback` yang mengambil input berupa skor akhir.

```
def generate_feedback(final_score):
    if final_score >= 90:
        feedback = "Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan. Masih ada ruang untuk perbaikan lebih lanjut. Berikut adalah beberapa saran berdasar"
    elif final_score >= 80:
        feedback = "Esai Anda cukup baik. Masih ada ruang untuk perbaikan lebih lanjut. Berikut adalah beberapa saran berdasarkan analisis mendalam."
    elif final_score >= 70:
        feedback = "Esai Anda sudah cukup baik, namun masih ada ruang untuk perbaikan lebih lanjut. Berikut adalah beberapa saran berdasarkan analisis mendalam."
    elif final_score >= 60:
        feedback = "Esai Anda masih membutuhkan perbaikan signifikan dalam penyampaian ide utama, struktur kalimat, dan koherensi."
    elif final_score <= 20:
        feedback = "Berikan jawaban esai anda sesuai dengan prompt yang diinginkan agar menghasilkan nilai yang baik"
    else:
        feedback = "Esai Anda masih kurang memenuhi kriteria yang diharapkan. Anda perlu meningkatkan penyampaian ide utama, struktur kalimat, dan koherensi."
    return feedback
```

**Gambar 4. 24** *Feedback Holistik*

```
def get_sentence_length_feedback(normalized_avg_sentence_length):
    if normalized_avg_sentence_length >= 0.8:
        return "Panjang kalimat Anda sangat baik dan menunjukkan kedalaman argumen yang bagus."
    elif normalized_avg_sentence_length >= 0.6:
        return "Panjang kalimat Anda cukup baik, tetapi bisa ditingkatkan sedikit lebih panjang untuk menambah kedalaman."
    elif normalized_avg_sentence_length >= 0.1:
        return "Panjang kalimat Anda terlalu pendek, sehingga argumen Anda tidak dapat dipahami"
    elif normalized_avg_sentence_length >= 0.0:
        return "Panjang kalimat Anda sangat pendek. Isi Essay Anda dengan baik dan benar"
```

**Gambar 4. 25** *Feedback Analitik*

Selain umpan balik berdasarkan skor akhir, sistem ini memberikan umpan balik sesuai dengan komponen-komponen penilaian yang diberikan. Hal ini bertujuan untuk memberikan masukan yang lebih spesifik kepada user dalam memberikan jawaban esai yang diminta.

#### 4.6.5 Final Score

Final skor merupakan skor akhir yang diberikan kepada jawaban esai yang dilakukan dengan menghitung skor gabungan dari *BERT score*, *text similarity* dan rubrik *score*.

```
# Menghitung skor akhir
combined_score = bert_score + text_similarity + rubric_score
print(f"Combined Score (bert_score + text_similarity + rubric_score): {combined_score}")

# Memberikan umpan balik berdasarkan skor akhir
final_score = combined_score / 3
feedback = generate_feedback(final_score)

return final_score, feedback
```

**Gambar 4. 26** *Final Score*

Final skor kemudian dihitung dengan membagi skor gabungan dengan 3 atau rata-rata dari ketiga skor. Skor akhir ini mencerminkan kualitas keseluruhan esai berdasarkan metrik-metrik yang digunakan.

## 4.9 Pengujian

Pengujian sistem ini dilakukan dengan menguji setiap soal yang ada dalam website sistem yang telah dibuat. Proses pengujian dilakukan dengan memberikan 5

jawaban untuk setiap soal yang ada. Setiap jawaban akan diberikan nilai atau kategori tertentu berdasarkan kualitasnya dan rentang nilai pada umpan balik. Berdasarkan umpan balik, ditentukan kategori seperti pada tabel 4.5 ini.

**Tabel 4. 5** Kategori Nilai

| Nilai  | Umpan Balik   | Kategori Nilai |
|--------|---|----------------|
| 90-100 | Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan   | Sangat Baik    |
| 80-89  | Esai Anda sudah baik  | Baik           |
| 70-79  | Esai cukup baik, namun masih ada ruang untuk perbaikan lebih lanjut | Cukup Baik     |
| 60-69  | Esai Anda masih membutuhkan perbaikan signifikan                    | Kurang Baik    |
| 59-0   | Esai Anda masih kurang memenuhi kriteria yang diharapkan.           | Sangat Kurang  |

Dengan kategori tersebut, dilakukan pengujian untuk setiap soal atau *prompt* dengan memberikan jawaban yang menghasilkan nilai, umpan balik dan kategori nilai yang telah diberikan.

#### **Pengujian Prompt Pertama**

Pengujian pertama dilakukan dengan meminta sistem menjawab soal "Tulis sebuah esai tentang pentingnya menjaga lingkungan" sebanyak 5 kali. Tujuannya adalah untuk mengevaluasi apakah sistem dapat memberikan penilaian yang konsisten dan akurat berdasarkan jawaban atau inputan yang diberikan oleh pengguna. Hasil pengujian untuk prompt atau soal pertama dapat dilihat pada tabel 4.6 berikut.

**Tabel 4. 6** Pengujian *Prompt* Pertama

| No | Jawaban User   | Nilai | Feedback  | Kategori   |
|----|--|-------|---|------------|
| 1. | Lingkungan yang bersih dan sehat sangat penting bagi kehidupan manusia. Dengan menjaga lingkungan, kita dapat mencegah berbagai masalah seperti pencemaran | 73    | Esai Anda cukup baik, namun masih ada ruang untuk | Cukup Baik |

|    |   |    |  |             |
|----|---|----|--|-------------|
|    | udara, tanah, dan air, yang dapat berdampak buruk pada kesehatan kita.  |    | perbaikan lebih lanjut.  |             |
| 2. | Secara keseluruhan, menjaga lingkungan adalah tanggung jawab bersama yang harus kita emban dengan serius. Kesehatan, kelestarian ekosistem, manfaat ekonomi, pendidikan, dan kebijakan yang tepat adalah elemen-elemen kunci dalam upaya pelestarian lingkungan. Dengan menjaga lingkungan, kita tidak hanya melindungi bumi tempat kita tinggal, tetapi juga menjamin kehidupan yang lebih baik bagi generasi mendatang. Oleh karena itu, mari kita mulai dari hal-hal kecil, seperti mengurangi penggunaan plastik, menanam pohon, dan membuang sampah pada tempatnya, untuk berkontribusi pada lingkungan yang lebih bersih dan sehat. | 87 | Esai Anda sudah baik.  | Baik        |
| 3. | Secara keseluruhan, menjaga lingkungan adalah tanggung jawab bersama yang harus kita emban dengan serius. Kesehatan, kelestarian ekosistem, manfaat ekonomi, pendidikan, dan kebijakan yang tepat adalah elemen-elemen kunci dalam upaya pelestarian lingkungan. Dengan menjaga lingkungan, kita tidak hanya melindungi bumi tempat kita tinggal, tetapi juga menjamin kehidupan yang lebih baik bagi generasi mendatang.   | 91 | Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan. | Sangat Baik |

|    |   |    |   |                    |
|----|---|----|---|--------------------|
|    | <p>Oleh karena itu, mari kita mulai dari hal-hal kecil, seperti mengurangi penggunaan plastik, menanam pohon, dan membuang sampah pada tempatnya, untuk berkontribusi pada lingkungan yang lebih bersih dan sehat. Selain itu, penting juga untuk mendorong inovasi dan pengembangan teknologi ramah lingkungan sebagai bagian dari solusi jangka panjang. Dengan terus berinvestasi dalam penelitian dan pembangunan infrastruktur yang berkelanjutan, kita dapat menciptakan masa depan yang lebih hijau dan berkelanjutan bagi semua. Contoh nyata dari tindakan ini adalah dengan mendukung program-program penanaman pohon lokal yang bertujuan untuk menghijaukan kota-kota kita dan meningkatkan kualitas udara. Kami juga dapat berpartisipasi dalam kampanye daur ulang untuk mengurangi limbah dan meningkatkan efisiensi sumber daya. Agar kita dapat menjaga lingkungan dengan baik, kita juga perlu melihat peran kita sebagai individu.</p> |    |   |                    |
| 4. | <p>Secara keseluruhan, menjaga lingkungan adalah tanggung jawab bersama yang harus kita emban dengan serius. Kesehatan, kelestarian ekosistem, manfaat ekonomi,</p>   | 65 | <p>Esai Anda masih membutuhkan perbaikan signifikan</p> | <p>Kurang Baik</p> |

|    |   |    |   |               |
|----|---|----|---|---------------|
|    | pendidikan, dan kebijakan yang tepat adalah elemen-elemen kunci dalam upaya pelestarian lingkungan. |    |   |               |
| 5. | dijaga terus saja agar baik   | 21 | Esai Anda masih kurang memenuhi kriteria yang diharapkan. | Sangat Kurang |

### Pengujian Prompt Kedua

Pengujian kedua dilakukan dengan meminta sistem menjawab soal "Machine Learning merupakan hal yang menjadi pembelajaran yang disukai mahasiswa, bisakah kamu menjelaskan kenapa machine learning menjadi kesukaan mahasiswa." sebanyak 5 kali. Tujuannya adalah untuk mengevaluasi apakah sistem dapat memberikan penilaian yang konsisten dan akurat berdasarkan jawaban atau inputan yang diberikan oleh pengguna. Hasil pengujian untuk *prompt* atau soal pertama dapat dilihat pada tabel 4.7 ini.

**Tabel 4. 7** Pengujian *Prompt* Kedua

| No. | Jawaban User  | Nilai | Feedback   | Kategori    |
|-----|---|-------|--|-------------|
| 1.  | Machine learning adalah cabang dari kecerdasan buatan di mana komputer belajar dari data yang diberikan untuk meningkatkan kinerjanya dalam menyelesaikan tugas tertentu tanpa perlu pemrograman eksplisit. Ini berbeda dari pendekatan konvensional di mana algoritma diprogram secara langsung untuk mengekstraksi informasi dari data. Proses utama dalam machine learning melibatkan persiapan data, pemilihan model, pelatihan model, evaluasi, dan iterasi. | 96    | Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan. | Sangat Baik |

|    |  |    |   |            |
|----|--|----|---|------------|
|    | <p>Persiapan data adalah langkah awal di mana data dikumpulkan, dibersihkan, dan diubah menjadi format yang sesuai untuk analisis. Selanjutnya, model yang sesuai dipilih berdasarkan jenis masalah yang ingin dipecahkan, seperti klasifikasi atau regresi. Setelah model dipilih, dilakukan pelatihan dengan menggunakan data yang telah disiapkan sebelumnya. Proses pelatihan ini mengharuskan model untuk menyesuaikan parameter internalnya agar dapat mengenali pola yang ada dalam data.</p> |    |   |            |
| 2. | <p>Machine learning adalah bagian dari kecerdasan buatan di mana komputer belajar dari data untuk meningkatkan kinerjanya tanpa pemrograman langsung. Berbeda dengan pendekatan tradisional, algoritma di sini tidak diprogram untuk mengekstraksi informasi secara eksplisit. Proses utamanya mencakup persiapan data, pemilihan model, pelatihan model, dan evaluasi.</p>  | 86 | Esai Anda sudah baik  | Baik       |
| 3. | <p>Machine learning menjadi kesukaan mahasiswa karena topiknya menarik dan sering dibicarakan. Banyak mahasiswa suka karena mereka merasa topik ini keren dan modern. Mereka juga mendengar bahwa banyak pekerjaan bagus membutuhkan pengetahuan tentang machine learning.</p>   | 73 | Esai cukup baik, namun masih ada ruang untuk perbaikan lebih lanjut | Cukup Baik |



|    |   |    |   |               |
|----|---|----|---|---------------|
|    | Selain itu, ada banyak sumber belajar online yang membuatnya lebih mudah dipelajari. Mahasiswa juga senang karena bisa mencoba membuat proyek kecil dengan machine learning dan melihat hasilnya dengan cepat.  |    |   |               |
| 4. | Machine learning disukai mahasiswa karena mereka suka. Topiknya populer dan banyak yang ngomongin. Banyak juga yang bilang kerjaan bagus butuh machine learning. Terus, banyak bahan belajar di internet jadi gampang. Mahasiswa juga suka bikin proyek kecil pakai machine learning karena cepat kelihatan hasilnya. | 63 | Esai Anda masih membutuhkan perbaikan signifikan          | Kurang Baik   |
| 5. | Machine learning disukai mahasiswa karena keren.  | 32 | Esai Anda masih kurang memenuhi kriteria yang diharapkan. | Sangat Kurang |

### Pengujian Prompt Ketiga

Dengan soal atau *prompt* “Jika CNN merupakan sebuah metode machine learning yang dikhususkan untuk data gambar, menurut kamu apakah CNN bisa digunakan dalam data teks ataupun suara, jelaskan.” dilakukan pengujian sebanyak 5 inputan user seperti pada tabel 4.8.

**Tabel 4. 8** Pengujian *Prompt* Ketiga

| No. | Jawaban User  | Nilai | Feedback             | Kategori |
|-----|---|-------|----------------------|----------|
| 1.  | Ya, CNN dapat digunakan untuk data teks dan suara meskipun awalnya dikembangkan untuk data gambar. Dalam pemrosesan teks, CNN dapat | 88    | Esai Anda sudah baik | Baik     |

|    |  |    |  |                    |
|----|--|----|--|--------------------|
|    | <p>menganalisis urutan kata atau karakter untuk mengenali pola seperti kata kunci atau frasa, yang berguna dalam klasifikasi teks atau analisis sentimen. Untuk suara, CNN dapat mengidentifikasi fitur-fitur audio dengan mengolah representasi spektrogram dari sinyal suara, memungkinkan pengenalan pola audio yang kompleks. Penyesuaian utama yang diperlukan adalah penggunaan lapisan konvolusi yang diadaptasi untuk data sekuensial atau temporal. Selain itu, kombinasi CNN dengan arsitektur lain seperti RNN (Recurrent Neural Network) atau LSTM (Long Short-Term Memory) dapat meningkatkan kinerja dalam pemrosesan teks dan suara dengan memanfaatkan kekuatan masing-masing metode untuk menangani urutan data dan kontekstual informasi. Dengan demikian, meskipun CNN awalnya dirancang untuk gambar, dengan penyesuaian yang tepat, CNN dapat menjadi alat yang sangat efektif untuk berbagai jenis data lain seperti teks dan suara.</p> |    |  |                    |
| 2. | <p>Ya, CNN dapat digunakan untuk data teks dan suara meskipun awalnya dikembangkan untuk data gambar. CNN mampu menganalisis teks dengan memproses urutan kata atau</p>  | 96 | <p>Esai Anda sangat baik dan memenuhi semua kriteria</p> | <p>Sangat Baik</p> |

|    |  |    |                        |            |
|----|--|----|------------------------|------------|
|    | <p>karakter, sehingga dapat mengenali pola-pola seperti kata kunci atau frasa, yang sangat berguna dalam tugas-tugas seperti klasifikasi teks dan analisis sentimen. Dalam konteks pemrosesan suara, CNN dapat mengidentifikasi fitur-fitur audio dengan mengolah representasi spektrogram dari sinyal suara, memungkinkan pengenalan pola audio yang kompleks. Penyesuaian yang diperlukan termasuk penggunaan lapisan konvolusi yang diadaptasi untuk data sekuensial atau temporal, sehingga CNN dapat menangani sifat berurutan dari data teks dan suara. Lebih jauh, menggabungkan CNN dengan arsitektur lain seperti RNN (Recurrent Neural Network) atau LSTM (Long Short-Term Memory) dapat memperkuat kinerja dalam pemrosesan teks dan suara, dengan memanfaatkan kemampuan masing-masing metode dalam menangani data berurutan dan informasi kontekstual. Dengan demikian, meskipun CNN awalnya dirancang untuk gambar, penyesuaian yang tepat memungkinkan CNN menjadi alat yang sangat efektif untuk berbagai jenis data lainnya seperti teks dan suara.</p> |    | yang diharapkan        |            |
| 3. | CNN adalah metode pembelajaran mesin yang biasanya dipakai untuk   | 78 | Esai cukup baik, namun | Cukup Baik |

|    |   |    |   |               |
|----|---|----|---|---------------|
|    | data gambar, tapi bisa juga digunakan untuk data teks atau suara, meskipun nggak begitu efektif. CNN lebih fokus ke ekstraksi fitur-fitur spasial, jadi untuk data teks atau suara yang lebih tentang urutan, mungkin metode lain seperti RNN lebih cocok.  |    | masih ada ruang untuk perbaikan lebih lanjut              |               |
| 4. | CNN bisa digunakan dalam data teks dan suara. Misalnya, kita bisa menggunakan CNN untuk menganalisis teks dengan merubah setiap kata menjadi vektor dan kemudian menerapkannya ke dalam lapisan-lapisan konvolusi. Begitu juga dengan suara, kita bisa mengubah sinyal suara menjadi representasi numerik | 67 | Esai Anda masih membutuhkan perbaikan signifikan          | Kurang Baik   |
| 5. | CNN bisa digunakan dalam data teks dan suara.   | 36 | Esai Anda masih kurang memenuhi kriteria yang diharapkan. | Sangat Kurang |

#### Pengujian Prompt Keempat

Prompt “Tuliskan esai yang menjelaskan peran penting deep learning dalam perkembangan teknologi modern. Jelaskan bagaimana deep learning telah mengubah paradigma di berbagai bidang, seperti pengenalan gambar, bahasa alami, dan pengolahan suara. Diskusikan juga tantangan dan potensi masa depan dalam pengembangan deep learning.” Hasil pengujian dapat dilihat pada tabel 4.9.

**Tabel 4. 9** Pengujian *Prompt* Keempat

| No. | Jawaban User | Nilai | Feedback | Kategori |
|-----|--------------|-------|----------|----------|
|-----|--------------|-------|----------|----------|

|    |   |    |   |             |
|----|---|----|---|-------------|
| 1. | Deep learning, terutama melalui Convolutional Neural Networks (CNN) untuk pengenalan gambar dan model seperti Transformer untuk pemrosesan bahasa alami, telah mengubah teknologi modern. CNN memungkinkan sistem mengenali objek dengan akurasi tinggi, diterapkan dalam kendaraan otonom dan pengenalan wajah. Sementara Transformer meningkatkan terjemahan otomatis dan chatbot yang lebih responsif. Namun, tantangan seperti interpretasi model dan ketersediaan data berkualitas tetap menjadi fokus pengembangan deep learning.   | 85 | Esai Anda sudah baik  | Baik        |
| 2. | Peran deep learning dalam era teknologi modern tidak dapat diabaikan. Dalam pengolahan suara, model seperti Recurrent Neural Networks (RNN) telah mengubah cara kita berinteraksi dengan perangkat, memungkinkan pengenalan suara yang lebih baik dan pengembangan asisten virtual yang lebih cerdas. Selain itu, dalam bidang kesehatan, deep learning telah memungkinkan deteksi penyakit dengan cepat dan akurat melalui analisis citra medis. Namun, penting untuk mengakui tantangan seperti keamanan dan privasi data dalam pemanfaatan teknologi deep learning ini. Deep learning, seperti Recurrent | 91 | Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan | Sangat Baik |

|    |   |    |   |               |
|----|---|----|---|---------------|
|    | Neural Networks (RNN), memberikan kontribusi besar dalam meningkatkan kualitas pengenalan suara dan diagnosis penyakit melalui citra medis, tetapi keberhasilannya juga menimbulkan tantangan baru terkait keamanan dan privasi data yang perlu diatasi secara serius. Meskipun demikian, dengan pengembangan yang tepat, deep learning memiliki potensi besar untuk terus membawa inovasi signifikan dalam berbagai bidang teknologi dan ilmu pengetahuan. |    |   |               |
| 3. | Deep learning merupakan teknologi yang sangat penting dalam perkembangan teknologi modern. Namun, masih ada tantangan yang perlu diatasi, seperti kebutuhan akan data yang besar dan berkualitas tinggi, serta interpretasi model yang kompleks.  | 77 | Esai cukup baik, namun masih ada ruang untuk perbaikan lebih lanjut | Cukup Baik    |
| 4. | Deep learning merupakan teknologi yang sangat penting dalam perkembangan teknologi modern. sehingga memerlukan pendalaman dalam mempelajari deep learning untuk meningkatkan teknologi.   | 62 | Esai Anda masih membutuhkan perbaikan signifikan                    | Kurang Baik   |
| 5. | saya tidak terlalu memahaminya, tetapi deep learning penting bagi perkembangan teknologi yang ada.  | 39 | Esai Anda masih kurang memenuhi kriteria yang diharapkan.           | Sangat Kurang |

### Pengujian Prompt Kelima

Pengujian prompt kelima untuk prompt “Tuliskan esai yang menjelaskan kegunaan dan signifikansi penggunaan BERT (Bidirectional Encoder Representations from Transformers) dalam mengolah data teks. Jelaskan bagaimana BERT telah meningkatkan pemahaman komputer terhadap konteks dan kompleksitas teks, serta dampaknya dalam berbagai aplikasi, seperti pemrosesan bahasa alami, analisis sentimen, dan pemahaman pertanyaan-jawaban.” Dilakukan sebanyak 5 kali pengujian seperti pada tabel 4.10 di bawah ini.

**Tabel 4. 10** Pengujian *Prompt* Kelima

| No. | Jawaban User  | Nilai | Feedback  | Kategori    |
|-----|---|-------|---|-------------|
| 1.  | BERT memainkan peran penting dalam pengolahan data teks modern. Dengan kemampuannya memahami konteks dan hubungan antar kata, BERT meningkatkan pemrosesan bahasa alami, meningkatkan analisis sentimen, dan memungkinkan pemahaman pertanyaan-jawaban yang lebih baik. Ini menghasilkan analisis yang lebih akurat dan mendalam, serta jawaban yang lebih relevan dan tepat. | 85    | Esai Anda sudah baik  | Baik        |
| 2.  | Penggunaan BERT (Bidirectional Encoder Representations from Transformers) telah merevolusi pemrosesan teks dengan meningkatkan pemahaman komputer terhadap konteks dan kompleksitas linguistik. Berbeda dengan model sebelumnya, BERT memahami kata-kata dalam konteks mereka yang sebenarnya, memperhitungkan hubungan antara kata-kata sebelum dan sesudahnya               | 94    | Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan | Sangat Baik |

|    |   |    |  |                   |
|----|---|----|--|-------------------|
|    | <p>dalam suatu kalimat. Ini berdampak besar dalam berbagai aplikasi, seperti pemrosesan bahasa alami, di mana BERT dapat menghasilkan representasi teks yang lebih akurat, meningkatkan kemampuan sistem untuk menjawab pertanyaan atau menerjemahkan bahasa. Dalam analisis sentimen, BERT mampu menangkap nuansa dan konteks emosional, memperbaiki keakuratan dalam memahami perasaan dalam teks. Dalam pemahaman pertanyaan-jawaban, BERT memungkinkan model untuk memahami pertanyaan secara lebih komprehensif, menghasilkan jawaban yang lebih relevan dan tepat. Dengan kemampuannya yang canggih, BERT telah membuka pintu bagi inovasi yang menakjubkan dalam pemrosesan teks dan aplikasi AI lainnya, membawa kita menuju era yang lebih cerdas dan responsif secara linguistik.</p> |    |  |                   |
| 3. | <p>BERT itu kayak alat yang bisa ngolah teks. Jadi, dia bantu komputer paham konteks dan teks yang ribet. Terus, aplikasinya luas banget, misalnya buat ngecek sentiment atau jawab pertanyaan. Jadi, pokoknya BERT ini bantu banget deh dalam bahasa alami dan komputer jadi lebih pintar gitu.</p>  | 72 | <p>Esai cukup baik, namun masih ada ruang untuk perbaikan lebih lanjut</p> | <p>Cukup Baik</p> |



|    |   |    |   |               |
|----|---|----|---|---------------|
| 4. | BERT itu kayak alat teks gitu lah. Dia bantu komputer paham konteks dan teks yang susah. Terus, bisa dipake buat ngecek sentiment atau jawab pertanyaan juga. Pokoknya, BERT ini kayak bikin komputer jadi lebih pintar aja gitu. | 66 | Esai Anda masih membutuhkan perbaikan signifikan          | Kurang Baik   |
| 5. | tidak ada hal yang bisa digunakan dalam BERT sih, jadi tidak ada yang perlu dijelaskan untuk kegunaannya.   | 16 | Esai Anda masih kurang memenuhi kriteria yang diharapkan. | Sangat Kurang |

Pada setiap pengujian yang dilakukan, dalam memberikan nilai pada setiap inputan user dilakukan tahapan penilaian melalui beberapa komponen evaluasi. Komponen tersebut meliputi skor *text similarity*, *BERT score*, dan skor penilaian berdasarkan rubrik. Setiap komponen memberikan kontribusi spesifik terhadap penilaian akhir. Proses perhitungan skor akhir menggunakan rumus sebagai berikut.

$$\text{Final score} = \frac{\text{Similarity score} + \text{BERT score} + \text{Rubric score}}{3}$$

Nilai akhir didapatkan dengan menjumlahkan skor dari masing-masing komponen kemudian dibagi dengan berapa banyak komponen yang ada, sehingga mencerminkan kualitas keseluruhan esai yang diuji.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil pengujian dan implementasi model BERT dalam melakukan penilaian esai, diperoleh kesimpulan sebagai berikut :

1. Dalam penelitian ini dataset dilatih sebanyak lebih kurang 5 kali latih hingga model BERT yang dilatih menggunakan *kappa score* mengalami peningkatan kinerja model dari *fold* ke *fold* hingga mendapatkan akurasi yang sempurna yaitu 1. Model yang dilatih juga mendapatkan hasil penilaian *kappa score* sebesar 0,82.
2. Untuk mendapatkan akurasi dan penilaian *kappa score* yang baik, diperlukan tahapan *pre-processing* dataset.
3. Word2Vec memegang peranan penting dalam pendekatan *text similarity* dengan menghasilkan beberapa representasi vektor dari sebuah esai sehingga memungkinkan analisis yang mendalam terhadap kesamaan dan perbedaan dalam konten tulisan tersebut.
4. Penggabungan *text similarity*, rubrik *score*, dan BERT *score* memiliki peran krusial dalam memberikan nilai akhir untuk prediksi skor esai, memastikan evaluasi yang holistic dan akurat terhadap kualitas tulisan siswa.
5. Model BERT yang dilatih sudah begitu baik dalam menilai jawaban esai yang memiliki jawaban yang panjang.
6. *Feedback* yang dibuat sistem hanya berdasarkan nilai akhir esai dan masih belum menggunakan model yang sesuai.

#### 5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambah dan menguji dataset Bahasa Indonesia yang lebih banyak untuk mendapatkan *kappa score* yang baik.
2. Disarankan untuk melakukan penelitian lebih lanjut dengan melakukan penilaian esai dengan jawaban matematis.
3. Disarankan melakukan penelitian untuk memberikan *feedback* atau umpan balik dengan menggunakan model yang sesuai.

4. Penelitian kedepannya sistem dapat diintegrasikan pada website pembelajaran mahasiswa.
5. Disarankan untuk melakukan lebih banyak *hyperparameter tuning* untuk perbandingan akurasi yang lebih baik
6. Disarankan untuk mengembangkan sistem agar dapat menggunakan *cloud* untuk sistem dapat menginputkan soal dan prompt esai secara publik dan *ter-update*.

## DAFTAR PUSTAKA

- Ali, Y. A., Awwad, E. M., Al-Razgan, M., & Maarouf, A. (2023). Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity. *Processes*, 11(2). <https://doi.org/10.3390/pr11020349>
- Badia, G. (2019). Holistic or analytic rubrics? Grading information literacy instruction. *College & Undergraduate Libraries*, 26(2), 109–116. <https://doi.org/10.1080/10691316.2019.1638081>
- Brownlee Disclaimer, J. (2017). *Deep Learning for Natural Language Processing Develop Deep Learning Models for Natural Language in Python Acknowledgements Copyright Deep Learning for Natural Language Processing*. →.
- Chen, H., & He, B. (2013). Automated essay scoring by maximizing human-machine agreement. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, October*, 1741–1752.
- Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.
- Djoko, E. R., Rikel, J., Mansor, M., & Slater, R. (2020). BERT for Question Answering on BioASQ. In *SMU Data Science Review* (Vol. 3, Issue 3). <https://scholar.smu.edu/datasciencereview> Available at: <https://scholar.smu.edu/datasciencereview/vol3/iss3/3http://digitalrepository.smu.edu>.
- Elalfi, A. E. E., Elgamal, A. F., & Amasha, N. A. (2019). Automated Essay Scoring using Word2vec and Support Vector Machine. In *International Journal of Computer Applications* (Vol. 177, Issue 25).
- Gomaa, W. H., & Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. In *International Journal of Computer Applications* (Vol. 68, Issue 13).
- Gunawansyah, Rahayu, R., Nurwathi, Sugiarto, B., & Gunawan. (2020, November 4). Automated essay scoring using natural language processing and text mining method. *Proceeding of 14th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2020*. <https://doi.org/10.1109/TSSA51342.2020.9310845>
- Halle, R. F., & Young, P. (2001). Analysis and Comparison of Heterogeneous. *Analysis*, June.
- 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>
- Hamner, B., Morgan, J., lynnvandev, Shermis, M., & Vander, T. (2012). *The Hewlett Foundation: Automated Essay Scoring*. Kaggle. <https://kaggle.com/competitions/asap-aes>
- 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>
- Keith, T. Z. (2003). Validity of Automated Essay Scoring Systems. In *Automated Essay Scoring: A Cross-Disciplinary Perspective* (Vol. 4, Issue 1). <https://doi.org/10.4324/9781410606860-19>
- 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>
- Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.
- Koroteev, M. (2021). *BERT: A Review of Applications in Natural Language Processing and Understanding*.
- 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>
- Lahitani, A. R. (2022). Automated Essay Scoring menggunakan Cosine Similarity pada Penilaian Esai Multi Soal. In *Jurnal Kajian Ilmiah* (Vol. 22, Issue 2). <http://ejurnal.ubharajaya.ac.id/index.php/JKI>
- Lipnevich, A. A., & Smith, J. K. (2009). Effects of Differential Feedback on Students' Examination Performance. *Journal of Experimental Psychology: Applied*, 15(4), 319–333. <https://doi.org/10.1037/a0017841>
- Lu, C., & Cutumisu, M. (2021). *Integrating Deep Learning into An Automated Feedback Generation System for Automated Essay Scoring*. <https://educationaldatamining.org/edm2021/>
- Lubis, F. F., Mutaqin, Putri, A., Waskita, D., Sulistyaningtyas, T., Arman, A. A., & Rosmansyah, Y. (2021). Automated Short-Answer Grading using Semantic Similarity based on Word Embedding. *International Journal of Technology*, 12(3), 571–581. <https://doi.org/10.14716/ijtech.v12i3.4651>
- Mattassi, R. (2015). Historical background. *Hemangiomas and Vascular Malformations: An Atlas of Diagnosis and Treatment*, 39–43. [https://doi.org/10.1007/978-88-470-5673-2\\_3](https://doi.org/10.1007/978-88-470-5673-2_3)
- McHugh, M. L. (2012). Lessons in biostatistics interrater reliability : the kappa statistic. *Biochemica Medica*, 22(3), 276–282. <https://hrcak.srce.hr/89395>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 1–9.
- N. Chamidah, M. M. Santoni, H. N. Irmanda, R. Astriratma, L. M. Tua, & T. Yuniati. (2021). *Word Expansion using Synonyms in Indonesian Short Essay Auto Scoring*.

- Noviani, Zubaidah, & Bistari. (2019). *PENGARUH UMPAN BALIK PEKERJAAN RUMAH TERHADAP RASA TANGGUNG JAWAB DAN HASIL BELAJAR MATEMATIKA*.
- Prasetya, D. D., Wibawa, A. P., & Hirashima, T. (2018). The performance of text similarity algorithms. *International Journal of Advances in Intelligent Informatics*, 4(1), 63–69. <https://doi.org/10.26555/ijain.v4i1.152>
- Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3), 2495–2527. <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*, 842–866. <https://doi.org/10.1162/tacl>
- Sadanand, V. S., Guruvyas, K. R. R., Patil, P. P., Acharya, J. J., & Suryakanth, S. G. (2022). An automated essay evaluation system using natural language processing and sentiment analysis. *International Journal of Electrical and Computer Engineering*, 12(6), 6585–6593. <https://doi.org/10.11591/ijece.v12i6.pp6585-6593>
- Schneider, J., & Vlachos, M. (2023). *A Survey of Deep Learning: From Activations to Transformers*.
- Sinu. (2022). *SISTEM PENILAIAN ESAI SECARA OTOMATIS UNTUK OPEN DOMAIN QUESTION ANSWERING DENGAN PENDEKATAN BERT (BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS)*.
- Tabassum, A., & Patil, R. R. (2020). A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing. *International Research Journal of Engineering and Technology*. [www.irjet.net](http://www.irjet.net)
- Universitas Amikom Yogyakarta, Universitas Gadjah Mada. Departemen Teknik Elektro dan Teknologi Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Purwokerto, Institute of Electrical and Electronics Engineers. Indonesia Section, & Institute of Electrical and Electronics Engineers. (n.d.). *ICITISEE 2018: the 3rd 2018 International conferences on Information Technology*
- Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023). *Attention Is All You Need*.
- Wang, J., & Dong, Y. (2020). Measurement of text similarity: A survey. In *Information (Switzerland)* (Vol. 11, Issue 9, pp. 1–17). MDPI AG. <https://doi.org/10.3390/info11090421>
- Wang, Y., Wang, C., Li, R., & Lin, H. (2022). *On the Use of BERT for Automated Essay Scoring: Joint Learning of Multi-Scale Essay Representation*. <http://arxiv.org/abs/2205.03835>
- Yang, L., & Shami, A. (2022). *On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice*.

## Lampiran

### Pengujian Website Nilai Cukup Baik

Soal: Tulis sebuah esai tentang pentingnya menjaga lingkungan.

Masukkan esai Anda:

Lingkungan yang bersih dan sehat sangat penting bagi kehidupan manusia. Dengan menjaga lingkungan, kita dapat mencegah berbagai masalah seperti pencemaran udara, tanah, dan air, yang dapat berdampak buruk pada kesehatan kita.

Get Score

Skor Akhir: 70.03924377759297

Umpan Balik Skor Akhir: Esai Anda sudah cukup baik, namun masih ada ruang untuk perbaikan lebih lanjut. Berikut adalah beberapa saran berdasarkan analisis mendalam

Umpan Balik Analitik:

Panjang kalimat Anda cukup baik, tetapi bisa ditingkatkan sedikit lebih panjang untuk menambah kedalaman.

Ide utama esai Anda tersampaikan dengan sangat baik.

Koherensi kalimat esai tidak dapat dipahami

## Pengujian Website Nilai Kurang Baik

digunakan dalam data teks ataupun suara, jelaskan.

Masukkan esai Anda:

CNN bisa digunakan dalam data teks dan suara. Misalnya, kita bisa menggunakan CNN untuk menganalisis teks dengan merubah setiap kata menjadi vektor dan kemudian menerapkannya ke dalam lapisan-lapisan konvolusi. Begitu juga dengan suara, kita bisa mengubah sinyal suara menjadi representasi numerik

Get Score

Skor Akhir: 67.6407121817271

Umpan Balik Skor Akhir: Esai Anda masih membutuhkan perbaikan signifikan dalam penyampaian ide utama, struktur kalimat, dan koherensi.

Umpan Balik Analitik:

Panjang kalimat Anda cukup baik, tetapi bisa ditingkatkan sedikit lebih panjang untuk menambah kedalaman.

Ide utama esai Anda cukup tersampaikan, tapi masih bisa lebih jelas.



## Pengujian Website Nilai Baik

Soal: Tulis sebuah esai tentang pentingnya menjaga lingkungan.

Masukkan esai Anda:

Secara keseluruhan, menjaga lingkungan adalah tanggung jawab bersama yang harus kita emban dengan serius. Kesehatan, kelestarian ekosistem, manfaat ekonomi, pendidikan, dan kebijakan yang tepat adalah elemen-elemen kunci dalam upaya pelestarian lingkungan. Dengan menjaga lingkungan, kita tidak hanya melindungi bumi tempat kita tinggal, tetapi juga menjamin kehidupan yang lebih baik bagi generasi mendatang. Oleh karena itu, mari kita mulai dari hal-hal kecil, seperti mengurangi penggunaan plastik, menanam pohon, dan membuang sampah pada tempatnya, untuk berkontribusi pada lingkungan yang lebih bersih dan sehat.

[Get Score](#)

Skor Akhir: 81.79784830411275

Umpan Balik Skor Akhir: Esai Anda sudah cukup baik. Masih ada ruang untuk perbaikan lebih lanjut. Berikut adalah beberapa saran berdasarkan analisis mendalam

Umpan Balik Analitik:

Panjang kalimat Anda sangat baik dan menunjukkan kedalaman argumen yang bagus.

Ide utama esai Anda tersampaikan dengan sangat baik.

Koherensi kalimat esai tidak dapat dipahami

## Pengujian Website Nilai Sangat Baik

### ESSAIGRADE

Soal: Jika CNN merupakan sebuah metode machine learning yang dikhususkan untuk data gambar, menurut kamu apakah CNN bisa digunakan dalam data teks ataupun suara, jelaskan.

Masukkan esai Anda:

Ya, CNN dapat digunakan untuk data teks dan suara meskipun awalnya dikembangkan untuk data gambar. CNN mampu menganalisis teks dengan memproses urutan kata atau karakter, sehingga dapat mengenali pola-pola seperti kata kunci atau frasa, yang sangat berguna dalam tugas-tugas seperti klasifikasi teks dan analisis sentimen. Dalam konteks pemrosesan suara, CNN dapat mengidentifikasi fitur-fitur audio dengan mengolah representasi spektrogram dari sinyal suara, memungkinkan pengenalan pola audio yang kompleks. Penyesuaian yang diperlukan termasuk penggunaan lapisan konvolusi yang diadaptasi untuk data sekuensial atau temporal, sehingga CNN dapat menangani sifat berurutan dari data teks dan suara. Lebih jauh, menghubungkan CNN

Get Score

Skor Akhir: 96.21598132451375

Umpan Balik Skor Akhir: Esai Anda sangat baik dan memenuhi semua kriteria yang diharapkan. Masih ada ruang untuk perbaikan lebih lanjut. Berikut adalah beberapa saran berdasarkan analisis mendalam

Umpan Balik Analitik:

## Pengujian Website Nilai Sangat Kurang

### ESSAI GRADER

Soal: Tuliskan esai yang menjelaskan peran penting deep learning dalam perkembangan teknologi modern. Jelaskan bagaimana deep learning telah mengubah paradigma di berbagai bidang, seperti pengenalan gambar, bahasa alami, dan pengolahan suara. Diskusikan juga tantangan dan potensi masa depan dalam pengembangan deep learning.

Masukkan esai Anda:

saya tidak terlalu memahaminya, tetapi deep learning penting bagi perkembangan teknologi yang ada.

Get Score

Skor Akhir: 39.400940577189125

Umpan Balik Skor Akhir: Esai Anda masih kurang memenuhi kriteria yang diharapkan. Anda perlu meningkatkan penyampaian ide utama, struktur kalimat, dan koherensi.