PENERAPAN *SUPERVISED FINE-TUNING* PADA *OPEN-SOURCE LARGE LANGUAGE MODEL* UNTUK PENILAIAN OTOMATIS ESAI BERBAHASA INDONESIA

(Skripsi)

Oleh

AHLAN SAYYID ALGHIFFARI
NPM. 2217051017

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

PENERAPAN *SUPERVISED FINE-TUNING* PADA *OPEN-SOURCE LARGE LANGUAGE MODEL* UNTUK PENILAIAN OTOMATIS ESAI BERBAHASA INDONESIA

Oleh

AHLAN SAYYID ALGHIFFARI

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRAK

PENERAPAN *SUPERVISED FINE-TUNING* PADA *OPEN-SOURCE LARGE LANGUAGE MODEL* UNTUK PENILAIAN OTOMATIS ESAI BERBAHASA INDONESIA

Oleh

AHLAN SAYYID ALGHIFFARI

Penilaian esai otomatis sangat penting untuk mengurangi subjektivitas serta beban kerja penilai manusia dalam dunia pendidikan. Penelitian ini bertujuan untuk mengimplementasikan *Supervised Fine-Tuning* (SFT) pada model *open-source Large Language Model* (LLM), yaitu Meta Llama 3.1 8B *Instruct*, untuk penilaian otomatis esai berbahasa Indonesia. Penelitian ini menggunakan desain eksperimen dengan memanfaatkan dataset UKARA 1.0 *Challenge* yang terdiri dari *Problem* A dan *Problem* B. Metodologi yang digunakan adalah *Parameter-Efficient Fine-Tuning* (PEFT) dengan pendekatan *4-bit Quantized Low-Rank Adaptation* (QLoRA) melalui *framework* Unsloth untuk mengatasi keterbatasan komputasi dan pemakaian memori. Empat skema data dievaluasi, yaitu data *Raw*, *Misspelling Correction* & *Normalization*, augmentasi *back-translation*, dan data kombinasi dari ketiga skema tersebut. Ketidakseimbangan kelas ditangani menggunakan pendekatan *weighted loss*, selain itu berbagai strategi *prompting* juga dilakukan pengujian. Hasil penelitian menunjukkan bahwa SFT secara signifikan meningkatkan performa dibandingkan pendekatan tanpa *fine-tuning*. Performa SFT terbaik diperoleh menggunakan data mentah dengan strategi *prompt template* menggunakan Llama *special tokens*, akurasi mencapai 89,93% pada *Problem* A dan 72,13% pada *Problem* B, dengan F1-*score* terbaik pada kelas positif masing-masing sebesar 93,16% dan 75,68%.

Kata kunci: Penilaian Otomatis Esai, SFT, LLM, QLoRA, Llama 3.1.

ABSTRACT

APPLICATION OF SUPERVISED FINE-TUNING ON AN OPEN-SOURCE LARGE LANGUAGE MODEL FOR AUTOMATED INDONESIAN ESSAY SCORING

By

AHLAN SAYYID ALGHIFFARI

Automated essay scoring plays a crucial role in reducing subjectivity and the workload of human raters in educational settings. This study aims to implement Supervised Fine-Tuning (SFT) on an open-source Large Language Model (LLM), namely Meta Llama 3.1 8B Instruct, for automated scoring of Indonesian-language essays. The research adopts an experimental design utilizing the UKARA 1.0 Challenge dataset, which consists of Problem A and Problem B. The methodology employs Parameter-Efficient Fine-Tuning (PEFT) using the 4-bit Quantized Low-Rank Adaptation (QLoRA) approach through the Unsloth framework to address computational and memory constraints. Four data schemes are evaluated: raw data, misspelling correction and normalization, back-translation augmentation, and a combination of these three schemes. Class imbalance is handled using a weighted loss approach, and various prompting strategies are also evaluated. The results indicate that SFT significantly improves performance compared to approaches without fine-tuning. The best SFT performance is achieved using raw data with a prompt template strategy utilizing Llama special tokens, achieving an accuracy of 89.93% on Problem A and 72.13% on Problem B, with the best F1-scores on the positive class reaching 93.16% and 75.68%, respectively.

***Keywords**: Automated Essay Scoring, SFT, LLM, QLoRA, Llama 3.1.*

Judul Skripsi	:	**PENERAPAN *SUPERVISED FINE-TUNING* PADA *OPEN-SOURCE LARGE LANGUAGE MODEL* UNTUK PENILAIAN OTOMATIS ESAI BERBAHASA INDONESIA**
Nama Mahasiswa	:	**Ahlan Sayyid Alghiffari**
Nomor Pokok Mahasiswa	:	2217051017
Program Studi	:	S1 Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Rahman Taufik, S.Pd., M.Kom.	**Riska Amalia Praptiwi, S.Kom., M.Cs.**
NIP. 19930627 202203 1 007	NIP. 19930702 202406 2 001

2. Mengetahui

Ketua Jurusan Ilmu Komputer	Ketua Program Studi
Dwi Sakethi, S.Si., M.Kom.	**Tristiyanto, M.I.S., Ph.D.**
NIP. 19680611 199802 1 001	NIP. 19810414 200501 1 001

MENGESAHKAN

1. Tim Penguji

   Ketua : **Rahman Taufik, S.Pd., M.Kom.** .....

   Sekretaris : **Riska Amalia Praptiwi, S.Kom., M.Cs.** .....

   Penguji Utama : **Tristiyanto, M.I.S., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: **9 April 2026**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ahlan Sayyid Alghiffari

NPM : 2217051017

Dengan ini menyatakan bahwa skripsi yang berjudul **"Penerapan *Supervised Fine-Tuning* pada *Open-Source Large Language Model* untuk Penilaian Esai Otomatis Berbahasa Indonesia"** merupakan hasil karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertulis pada skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari ditemukin bukti bahwa skripsi saya merupakan hasil penjiplakan atau dibuat oleh orang lain maka saya bersedia menerima sanksi sesuai hukum yang berlaku.

Bandar Lampung, 23 April 2026
Penulis.

Ahlan Sayyid Alghiffari
NPM. 2217051017

RIWAYAT HIDUP

Penulis dilahirkan di Kotabumi Lampung Utara pada tanggal 7 September 2004. Penulis merupakan anak terakhir dari tiga bersaudara. Penulis menyelesaikan pendidikan dasar di Madrasah Ibtidaiyah Negeri 1 Lampung Utara pada Tahun 2016, kemudian menyelesaikan pendidikan di SMP Negeri 7 Kotabumi pada Tahun 2019 dan SMA Negeri 3 Kotabumi pada Tahun 2022.

Pada Tahun 2022 penulis terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer Universitas Lampung melalui jalur SNMPTN. Selama menempuh studi di perguruan tinggi, penulis terlibat dalam beberapa kegiatan sebagai berikut.

1. Asisten Dosen pada Mata Kuliah Logika Program Studi S1 Ilmu Komputer Universitas Lampung Tahun 2022.
2. Anggota Bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer Tahun 2023.
3. Peserta Studi Independen Bersertifikat di PT Dicoding Akademi Indonesia melalui program *Bangkit Academy Batch* 2 dengan jalur pembelajaran *Machine Learning* Tahun 2024.
4. Asisten Dosen pada Mata Kuliah Kecerdasan Buatan Program Studi S1 Ilmu Komputer Universitas Lampung Tahun 2025.
5. Peserta Magang Mandiri di PT Bukit Asam Tbk Unit Pelabuhan Tarahan Tahun 2025.
6. Peserta Kuliah Kerja Nyata (KKN) di Desa Wates Kecamatan Bumi Ratu Nuban Kabupaten Lampung Tengah Provinsi Lampung Periode I Tahun 2025.

MOTTO

"*Berniaga dengan Tuhan*
tidak mendatangkan kerugian"

PERSEMBAHAN

Alhamdulillahi Robbil 'aalamiin.

Atas nama Allah Subhanahu wa Ta'ala, dengan segala rahmat dan rida-Nya, sehingga skripsi ini dapat diselesaikan dengan baik. Selawat dan salam semoga senantiasa tercurah kepada Rasulullah Muhammad Shallallahu 'Alaihi wa Sallam.

Skripsi ini di persembahkan kepada:

Orang tua, keluarga, dan seluruh pihak yang telah memberikan doa, dukungan dan motivasi selama penyusunan skripsi ini, serta kepada diri penulis sebagai bentuk pencapaian dalam menyelesaikan pendidikan pada Program Studi S1 Ilmu Komputer Universitas Lampung.

SANCAWACANA

Puji syukur kehadirat Allah SWT atas segala rahmat, nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "**Penerapan *Supervised Fine-Tuning* pada *Open-Source Large Language Model* untuk Penilaian Otomatis Esai Berbahasa Indonesia**" dengan baik. Skripsi ini disusun sebagai salah satu syarat akademik utama untuk memperoleh gelar sarjana pada Program Studi S1 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Dengan penuh kerendahan hati pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Orang tua penulis yang senantiasa memberikan doa, dukungan dan kasih sayang serta kesempatan kepada penulis untuk menempuh pendidikan hingga selesai. Ucapan terima kasih juga disampaikan kepada kedua kakak penulis atas doa dan motivasi yang diberikan.
2. Bapak Dr. Eng. Heri Satria, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
3. Bapak Dwi Sakethi, S.Si., M.Kom., selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
4. Ibu Yunda Heningtyas, M.Kom., selaku Sekretaris Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D., selaku Ketua Program Studi S1 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung sekaligus Dosen Pembahas yang telah memberikan kritik dan saran yang membangun dalam proses penelitian dan penulisan skripsi ini.

6. Prof. Dr. Eng. Admi Syarif, selaku Dosen Pembimbing Akademik yang telah memberikan bimbingan dan arahan selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Bapak Rahman Taufik, S.Pd., M.Kom., selaku Dosen Pembimbing Utama skripsi yang telah memberikan kesempatan, arahan serta bimbingan kepada penulis sehingga penelitian ini dapat terselesaikan dengan baik.
8. Ibu Riska Amalia Praptiwi, S.Kom., M.Cs., selaku Dosen Pembimbing Kedua skripsi yang telah memberikan bimbingan, arahan serta motivasi yang bermanfaat selama proses penyusunan skripsi ini.
9. Bapak/Ibu Dosen serta seluruh staf Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan ilmu pengetahuan dan membantu dalam berbagai keperluan akademik maupun administratif.
10. Seluruh teman-teman yang telah memberikan dukungan, kebersamaan serta berbagai kenangan dan pengalaman berharga selama masa perkuliahan hingga proses penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih belum sempurna dan terdapat berbagai kekurangan. Oleh karena itu, penulis terbuka terhadap kritik dan saran yang membangun. Diharapkan skripsi ini dapat memberikan manfaat bagi berbagai pihak.

Bandar Lampung, 23 April 2026

Ahlan Sayyid Alghiffari
NPM. 2217051017

DAFTAR ISI

Halaman

DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv

I. PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 3
1.3. Batasan Masalah 4
1.4. Tujuan Penelitian 4
1.5. Manfaat Penelitian 5

II. TINJAUAN PUSTAKA 6
2.1. Penelitian Terdahulu 6
2.2. Penilaian Otomatis Esai (*Automatic Essay Scoring*) 11
2.3. *Natural Language Processing* (NLP) 13
2.4. *Large Language Models* (LLMs) 14
2.4.1. LLaMa 3 *Open-Source* LLM 16
2.5. Prapemrosesan Teks (*Text Preprocessing*) 17
2.5.1. *Data Cleaning* 18
2.5.2. Normalisasi Data 19
2.5.3. *Misspelling Correction* 19
2.5.4. Augmentasi Data 20
2.5.5. *Prompt Templating* 20
2.6. Tokenisasi 22
2.7. *Fine-Tuning* 23

2.7.1. Data *Imbalance Handling* 24
2.7.2. *Supervised Fine-Tuning* 25
2.7.3. *Hyperparameter* 26
2.7.4. *Parameter-Efficient Fine-Tuning* (PEFT) 28
2.7.5. Hugging Face 33
2.7.6. Unsloth 33
2.8. *Confusion Matrix* 34
2.8.1. Akurasi 35
2.8.2. Presisi 35
2.8.3. *Recall* 36
2.8.4. F1-*score* 36

III. METODOLOGI PENELITIAN 37
3.1. Tempat dan Waktu Penelitian 37
3.1.1. Tempat Penelitian 37
3.1.2. Waktu Penelitian 37
3.2. Perangkat Penelitian 38
3.2.1. Perangkat Keras 38
3.2.2. Perangkat Lunak 38
3.3. Tahap Penelitian 39
3.3.1. Pengumpulan Dataset 39
3.3.2. *Data Preprocessing* 41
3.3.3. *Model Preparation* 46
3.3.4. Tokenisasi 47
3.3.5. *Supervised Fine-Tuning* 48
3.3.6. Evaluasi Model 49

V. SIMPULAN DAN SARAN 50
5.1. Simpulan 50
5.2. Saran 51

DAFTAR PUSTAKA 52

DAFTAR TABEL

Tabel Halaman

1. Penelitian terdahulu 6
2. *Confusion Matrix* 35
3. Rencana penelitian 37
4. Spesifikasi perangkat keras 38
5. Sampel data UKARA *Problem* A 41
6. Sampel data UKARA *Problem* B 41
7. Contoh penerapan teks *Cleaning* 42
8. Contoh penerapan *Misspelling Correction* 43
9. Contoh penerapan normalisasi teks 43
10. Contoh penerapan *Back Translation* 44
11. *Hyperparameter* pada LoRA *Adapter* 47
12. *Hyperparameter Supervised Fine-Tuning* model 49

DAFTAR GAMBAR

Gambar Halaman

1. Ekosistem *Large Language Model* (Vaniukov, 2023). 14
2. Arsitektur *Transformer* (Vaswani *et al.*, 2017). 15
3. Proses *Prompt Templating*. 21
4. Alur *Prompt, Tokenizer,* dan *Embedding* (Tamang, 2024). 22
5. Ilustrasi proses *Fine-Tuning* LLM. 23
6. Ilustrasi implementasi *Supervised Fine-Tuning* (Huizenga *and* Hu, 2024). 25
7. Alur proses implementasi *Parameter Efficient Fine-Tuning*. 28
8. Ilustrasi *Low-Rank Adaptation* (Hu *et al.*, 2021). 30
9. Ilustrasi proses *Decomposition* matriks pada LoRA. 31
10. Ilustrasi *Quantized Low-Rank Adaptation* (Dettmers *et al.*, 2023). 32
11. Diagram alur penelitian. 40
12. *Prompt Templating* pada tiap skema dataset. 45
13. *Alur pelatihan model Transformer* (Tunstall *et al.*, 2022). 47
14. Ilustrasi proses tokenisasi. 48

I. PENDAHULUAN

1.1. Latar Belakang

Asesmen memegang peran penting dalam proses pembelajaran untuk mengevaluasi pencapaian dan kemampuan siswa. Ramesh dan Sanampudi (2022) menjelaskan bahwa asesmen membantu pendidik dalam memetakan pemahaman materi serta pencapaian target pembelajaran. Esai sebagai salah satu instrumen penilaian, mengukur kemampuan siswa secara tertulis dalam menganalisis, mengorganisasi, dan mengekspresikan ide (Pradani dan Suadaa, 2023). Meskipun demikian, metode ini memiliki kelemahan utama pada aspek subjektivitas karena sering menyebabkan ketidaksesuaian hasil evaluasi di antara para penilai (Awidi, 2024). Perbedaan standar dan interpretasi tersebut berisiko menciptakan ketidakadilan dalam proses penilaian (Kotha *et al.*, 2023). Selain itu kondisi internal penilai seperti kelelahan atau keterbatasan ingatan sering kali menyebabkan hasil penilaian menjadi tidak konsisten, di mana jawaban yang sama bisa mendapatkan nilai yang berbeda (Pradani dan Suadaa, 2023). Kompleksitas inilah yang mengharuskan penilai memiliki konsistensi dan kemampuan analisis tinggi dalam mengevaluasi substansi esai.

Berbagai kendala tersebut mendorong pemanfaatan teknologi sebagai solusi alternatif, salah satunya melalui sistem penilaian otomatis esai atau *Automatic Essay Scoring (*AES*)* yang dikembangkan sebagai teknologi untuk mengevaluasi dan memberi skor pada esai tertulis (Mizumoto *and* Eguchi, 2023). Penerapan AES tidak hanya bertujuan untuk mempercepat proses penilaian, tetapi juga untuk meningkatkan konsistensi, mengurangi bias penilai, serta mengoptimalkan efisiensi dalam konteks pendidikan maupun

asesmen berskala besar. AES menjadi sebuah solusi untuk mengatasi masalah waktu, biaya, dan beban kerja guru atau *evaluator* yang berlebih dalam melakukan penilaian esai secara manual (Lim *et al.*, 2021).

Seiring perkembangan penelitian AES, berbagai studi mulai menggunakan dataset relevan dalam bahasa tertentu. Dalam bahasa Indonesia, dataset yang banyak dikaji adalah UKARA 1.0 *Challenge*, kumpulan jawaban esai siswa yang dikembangkan tim *Natural Language Processing* Universitas Gadjah Mada bersama PUSPENDIK tahun 2019 (Universitas Gadjah Mada, 2019). Beragam pendekatan *Transformer* dan *Deep Learning* terbukti meningkatkan akurasi pada dataset ini, seperti studi oleh Tanaka dan peneliti lainnya (2024) mengombinasikan *Neural Network Classifier* dan BERT *embedding*, serta penelitian oleh Fadilah dan Priyanta (2024) yang mengombinasikan BiLSTM dengan IndoBERT *embedding*. Meskipun masih terbatas dalam memahami konteks dan kompleksitas penilaian esai, pendekatan ini sejalan dengan tren model *Transformer* yang unggul dibanding pendekatan konvensional.

Peningkatan kinerja sistem AES telah dicapai, namun menurut Pack, Barrett, dan Escalante (2024) penilaian yang diberikan oleh manusia dan mesin masih belum selaras. Ketidakselarasan ini utamanya timbul karena kriteria atau aspek yang dinilai oleh keduanya berbeda. Penelitian dengan dataset UKARA sebelumnya oleh Tanaka dan beberapa peneliti lainnya (2024) berjudul "*Evaluation of Back Translation and Misspelling Correction Utilization on Indonesian AES*" menyoroti bahwa model *Transformer* belum sepenuhnya menangkap nuansa semantik esai siswa. Studi tersebut merekomendasikan eksplorasi *Large Language Model* (LLM) dalam AES mengingat kemampuan *reasoning* LLM yang lebih baik dan pengetahuan internal yang luas, sementara pendekatan pada studi-studi sebelumnya masih terbatas pada model tradisional.

Sebagai respons terhadap rekomendasi tersebut, LLM muncul sebagai paradigma baru yang lebih efektif dan menjanjikan terutama untuk tugas seperti AES (Pack *et al.*, 2024; Song *et al.*, 2024) dengan kemampuan memahami konteks, struktur, dan koherensi teks lebih mendalam, LLM

menawarkan peningkatan signifikan dibanding model tradisional. Namun demikian, model berskala besar seperti GPT (*Generative Pre-trained Transformer*) memerlukan komputasi tinggi dan biaya implementasi mahal (Ormerod *et al.*, 2021). Oleh karena itu, penelitian ini memanfaatkan LLM *open-source* sebagai alternatif yang dinilai mampu mengatasi sebagian besar keterbatasan model *closed-source*, tanpa mengorbankan kinerja dalam tugas penilaian esai otomatis (Chen *et al.*, 2024).

Dalam konteks penelitian ini, karakteristik penilaian pada dataset UKARA 1.0 *Challenge* menuntut kemampuan model dalam memahami konteks linguistik, struktur esai, serta variasi ekspresi bahasa Indonesia secara komprehensif. Kondisi tersebut menjadikan LLM *open-source* berpotensi memainkan peran yang lebih signifikan dalam asesmen bahasa, khususnya apabila model disesuaikan (*fine-tuned*) untuk kebutuhan penilaian yang spesifik (Pack *et al.*, 2024), untuk mengoptimalkan kemampuan tersebut, diperlukan *Supervised Fine-Tuning* (SFT), yaitu pelatihan tambahan yang memungkinkan model beradaptasi dengan instruksi manusia dan karakteristik tugas tertentu (Dong *et al.*, 2024). Pendekatan ini menjadi dasar penelitian dalam penerapan LLM *open-source* untuk penilaian otomatis esai berbahasa Indonesia secara lebih efektif.

Berdasarkan uraian latar belakang tersebut maka penulis menetapkan untuk melakukan penelitian dengan judul "**Penerapan *Supervised Fine-Tuning* pada *Open-Source Large Language Model* untuk Penilaian Otomatis Esai Berbahasa Indonesia**".

1.2. Rumusan Masalah

Berdasarkan pemaparan pada latar belakang maka rumusan masalah dalam penelitian ini adalah bagaimana menerapkan *Supervised Fine-Tuning* pada *open-source Large Language Model* untuk melakukan penilaian otomatis esai berbahasa Indonesia.

1.3. Batasan Masalah

Agar penelitian ini lebih terarah dan fokus pada tujuan utama, maka batasan masalah yang ditetapkan adalah sebagai berikut:

1. Penelitian ini menggunakan dataset UKARA 1.0 Challenge dengan total 2.861 sampel yang terbagi menjadi dua jenis masalah. Dataset ini telah disediakan dalam bentuk pemisahan data latih, data validasi, dan data uji, sehingga tidak diperlukan proses pembagian data (*data split*) tambahan.
2. Penelitian ini terbatas pada evaluasi kinerja model *Open-source Large Language Model* yaitu *Meta-Llama-3.1-8B-Instruct* (versi Unsloth) sebagai model dasar yang diadaptasi, tanpa membandingkan dengan model lain.
3. Proses *fine-tuning* dioptimalkan menggunakan *framework* Unsloth dengan pendekatan QLoRA *4-bit* yang dipercepat oleh kernel Triton bawaan Unsloth, untuk mengatasi keterbatasan sumber daya komputasi.
4. Metode pelatihan yang digunakan adalah *Supervised Fine-Tuning* (SFT) tanpa melakukan perbandingan dengan metode pelatihan lainnya.
5. Penelitian ini berfokus pada tahap implementasi dan evaluasi model dalam konteks penilaian otomatis esai, tanpa mencakup penerapan langsung dalam sistem asesmen pendidikan nyata. Oleh karena itu, hasil penelitian ini bersifat awal dan memerlukan validasi lebih lanjut sebelum diterapkan secara praktis.

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan *Supervised Fine-Tuning* pada *open-source* LLM untuk penilaian otomatis esai berbahasa Indonesia.
2. Mengevaluasi kinerja model hasil *fine-tuning* menggunakan dataset UKARA 1.0 *Challenge*, untuk menilai kemampuan model dalam melakukan penilaian otomatis pada esai.

3. Membandingkan kinerja model dengan penelitian sebelumnya yang menggunakan dataset UKARA 1.0 *Challenge*.
4. Mengidentifikasi faktor-faktor yang memengaruhi kinerja model, meliputi variasi penggunaan dataset, hingga parameter pelatihan yang diterapkan.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan kontribusi terhadap pengembangan penerapan *Supervised Fine-Tuning* pada *open-source Large Language Model* untuk penilaian otomatis esai berbahasa Indonesia.
2. Memberikan wawasan tentang bagaimana *Supervised Fine-Tuning* pada *Large Language Model* dapat diterapkan dalam tugas penilaian esai berbahasa Indonesia.
3. Menjadi referensi bagi penelitian selanjutnya dalam pengembangan tugas *Automatic Essay Scoring* (AES) berbasis *Large Language Model* yang lebih adaptif dan efisien.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian yang pernah dilakukan terkait penilaian esai otomatis menggunakan dataset UKARA 1.0 *Challenge* dapat dilihat pada Tabel 1, sebagai berikut.

Tabel 1. Penelitian terdahulu

Peneliti	Judul	Metode	Hasil
Ilham Firdausi Putra (2020) (Septiandri dkk., 2020)	*Indonesian Essay Scoring using* Bi-LSTM *with Word Embedding Representation*	*Bidirectional Long Short-Term Memory* (Bi-LSTM) dengan *pretrained Word2Vec* dan validasi *Repeated Stratified K-Fold*.	Penelitian tersebut menghasilkan F1-*score* gabungan sebesar 81%, dengan *Problem* A 89,2% dan *Problem* B 77%.
Rian Adam Rajagede dan Rochana Prih Hastuti (Rajagede *and* Hastuti, 2020, 2021)	*Stacking Neural Network Models for Automatic Short Answer Scoring/ Automatic Short Answer Scoring for Bahasa Indonesia with Classifier Stacking*	*Stacking Multi-Layer Perceptron* (MLP) + *XGBoost*, dengan *FastText Embedding*, SMOTE, dan optimasi *hyperparameter* (*Optuna*-TPE).	Penelitian tersebut berhasil mencapai F1-*Score* gabungan 82,1%, dengan *Problem* A 88,3% dan *Problem* B 75,8%.

Tabel 1. (Lanjutan)

Peneliti	Judul	Metode	Hasil
Ali Akbar Septiandri, Yosef Ardhito Winatmoko (Septiandri *and* Winatmoko, 2020)	*UKARA 1.0 Challenge Track 1: Automatic Short-Answer Scoring in Bahasa Indonesia*	*Random Forest* dengan *Unigram* + LSA untuk *Problem A* dan *Logistic Regression* dengan TF-IDF untuk *Problem B*, melibatkan *lemmatization*, *Normalization*, serta validasi 5-*fold*.	Penelitian tersebut menghasilkan F1-*score* gabungan 81,2%, dengan *Problem* A 87,9% dan *Problem* B 76,4%.
Rian Adam Rajagede (Rajagede, 2021)	*Improving automatic essay scoring for Indonesian language using simpler model and richer feature*	*Neural Network* dengan *Sentence* BERT *Embedding*, *dropout*, SMOTE, dan optimasi *hyperparameter* (Optuna-*TPE*).	Penelitian tersebut berhasil mencapai F1-*score* gabungan 82,9%, dengan *Problem* A 89,4% dan *Problem* B 75,7%.
Nur Fadilah dan Sigit Priyanta (Fadilah *and* Priyanta, 2022)	*Automatic Essay Scoring Using Data Augmentation in Bahasa Indonesia*	Bi-LSTM dengan *FastText embedding* dan EDA (*Easy Data Augmentation*) menggunakan IndoBERT *embedding*, dan validasi *K-Fold*.	Penelitian tersebut mencapai Akurasi untuk *Problem* A 84,81% (Tanpa Augmentasi, dengan *K-fold*) dan *Problem* B 72,50% (EDA *Insert*, dengan *K-fold*)
Elvina Amadea Tanaka *et al* (Tanaka *et al.*, 2024)	*Evaluating Back Translation and Misspelling Correction Utilization on Indonesian AES*	*Neural Network* (1–3 *hidden layers*) dengan SBERT*, Back Translation,* koreksi ejaan *Norvig*, SMOTE, dan optimasi *hyperparameter* (*Optuna*-TPE).	Penelitian tersebut menghasilkan F1-*score* gabungan 83,4%, dengan *Problem* A 89,7% dan *Problem* B 77,2%.

Tabel 1. (Lanjutan)

Peneliti	**Judul**	**Metode**	**Hasil**
Ramadhania Humaira (Humaira, 2025)	*Automated Essay Scoring untuk Penilaian Jawaban Esai Bahasa Indonesia dengan IndoBERT Embedding dan Feedforward Neural Network*	*Feedforward Neural Network* dengan IndoBERT *embedding*, SMOTE, penambahan *distance feature*, dan optimasi hyperparameter.	Penelitian tersebut menghasilkan F1-*score* gabungan 76,7%, dengan *Problem* A 83,5% dan *Problem* B 69,9%.

Dataset UKARA 1.0 *Challenge* menjadi korpus yang dipakai dalam sejumlah penelitian untuk tugas *Automatic Short Answer Scoring* (penilaian jawaban singkat) atau *Automatic Essay Scoring* (AES) berbahasa Indonesia. Dataset ini dipublikasikan oleh *NLP Research Group Universitas Gadjah Mada* (UGM) pada tahun 2019 sebagai dataset pada UKARA *Challenge* 1.0 (Universitas Gadjah Mada, 2019).

Penelitian oleh Putra (2020), penelitian ini memperoleh peringkat kedua pada kompetisi UKARA 1.0 dengan menerapkan arsitektur *Bi-directional Long Short-Term Memory* untuk tugas klasifikasi biner. Sebagai representasi kata, peneliti menggunakan *Word2Vec* berukuran 100 dimensi yang bersifat *pretrained*. Untuk memperkaya cakupan kosakata model *embedding*, *Word2Vec* tersebut dilatih menggunakan gabungan korpus (Wikipedia Indonesia, korpus Opensubs, serta dataset UKARA). Tahap prapemrosesan relatif sederhana, meliputi *lowercasing* dan *remove punctuation*. Pada prosedur eksperimen, peneliti memakai *Repeated Stratified K-Fold* dengan 10 *split* dan 10 pengulangan, sehingga membentuk ensemble beranggotakan 100 model. Evaluasi pada *test set* menghasilkan F1-*score* 81%.

Penelitian Rajagede dan Hastuti (2020) mengusulkan pendekatan *stacking* yang mengombinasikan beberapa model dasar, yaitu dua *Artificial Neural Network* sederhana (*Multi-layer Perceptron*/MLP) dan XGBoost, dengan sebuah MLP sederhana bertindak sebagai *meta-classifier* pada lapisan akhir *stacking*. Fitur utama yang dipakai adalah FastText *sentence embedding* pada dua ukuran dimensi (rd300 dan rd150), yang kemudian dikombinasikan dengan fitur numerik berupa jumlah kata pada respons. Tahap prapemrosesan mencakup *case folding* dan penghapusan kata-kata yang sering muncul (*Stopwords*). Untuk optimasi *hyperparameter*, peneliti menggunakan algoritma *Tree-Structured Parzen Estimator* (TPE). Karena adanya ketidakseimbangan kelas, diterapkan teknik SMOTE untuk melakukan *upsampling* terhadap kelas minoritas sebesar 10%. Hasil akhir menunjukkan model *stacking* terbaik mencapai *Combined* F1-*score* 82,1%, mengungguli metode lain yang dipublikasikan pada waktu tersebut.

Penelitian Septiandri dan Winatmoko (2020) menempuh strategi berbeda dengan menggunakan model tunggal yang disesuaikan untuk tiap soal, yaitu *Random Forest* digunakan untuk soal A, sedangkan *Logistic Regression* dipilih untuk soal B. Proses ekstraksi fitur bervariasi sesuai karakteristik soal. Pada soal A peneliti memakai unigram yang diperkaya lewat *Latent Semantic Analysis* (LSA), sementara pada soal B digunakan fitur TF-IDF. Tahap prapemrosesan meliputi tokenisasi dan *lemmatization*. Selain itu, peneliti melakukan eksperimen dengan melakukan perbaikan manual terhadap label pada *training-set* untuk mencoba mengurangi inkonsistensi label yang terdeteksi. Pelatihan model dilaksanakan menggunakan *5-fold cross validation* pada data latih. Model terbaik yang diperoleh memiliki F1-*score* keseluruhan 81,2%. Penulis juga mencatat kemungkinan adanya *noise* pada *test-set*, sehingga perbaikan label manual pada *training set* tidak selalu merepresentasikan kondisi label pada *test-set*.

Penelitian lanjutan oleh Rajagede (2021) mengajukan pendekatan yang lebih sederhana pada arsitektur model: sebuah *Neural Network* (NN) dengan satu

lapisan tersembunyi. Perbedaan penting pada studi ini adalah penggunaan SBERT (*Sentence*-BERT) *sentence embedding* sebagai fitur yang dianggap lebih kaya representasinya dibanding *embedding* konvensional. Untuk penanganan dataset, dilakukan prapemrosesan termasuk penghapusan kata-kata yang sering muncul. Selain itu, penelitian ini menerapkan SMOTE dengan peningkatan kelas minoritas sebesar 10% untuk mengatasi ketidakseimbangan label. Optimasi *hyperparameter* dilakukan menggunakan Optuna dengan algoritma TPE, dan peneliti juga menerapkan teknik pencegahan *overfitting* seperti *dropout* serta strategi *incremental batch size*. Hasil eksperimen menunjukkan bahwa model NN sederhana ini berhasil mencapai F1-*score* 82,9%, yang menandai peningkatan dibandingkan laporan kinerja terbaik sebelumnya 82,1%.

Penelitian Fadilah dan Priyanta (2022) berfokus pada eksplorasi teknik augmentasi data melalui metode *Easy Data Augmentation* (EDA) pada dataset UKARA. Empat teknik EDA yang dieksplorasi adalah: *Synonym Replacement* (SR) (dimodifikasi dengan memanfaatkan *embedding* IndoBERT untuk pemilihan sinonim), *Random Insertion* (RI), *Random Swap* (RS), dan *Random Deletion* (RD). Model klasifikasi yang digunakan adalah BiLSTM dilengkapi *embedding* FastText. Dalam prosedur eksperimen, data latih dan validasi digabungkan terlebih dahulu sebelum dilakukan augmentasi, untuk kemudian menerapkan *k-fold cross validation*. Penulis menemukan bahwa augmentasi EDA yang dilakukan tanpa mempertimbangkan distribusi label tetap tidak mampu menyeimbangkan dataset (augmentasi bersifat label-agnostic). Hasil empiris menunjukkan bahwa pada soal A, kinerja tertinggi (akurasi 85,07%) diperoleh ketika menggunakan data tanpa augmentasi pada skema *k-fold cross validation*. Sebaliknya, pada soal B, teknik *Random Insertion* yang diaplikasikan dalam skema *k-fold cross validation* memberikan akurasi tertinggi (72,78%).

Tanaka *et al*. (2024) berhasil meningkatkan performa khususnya pada soal B dari dataset UKARA melalui penggunaan SBERT *sentence embedding* serta

model *Neural Network* (NN), dengan perhatian utama pada penyempurnaan tahapan prapemrosesan. Dua teknik augmentasi yang diuji adalah *Back Translation* (BT) dan *Misspelling Correction* (MC) menggunakan algoritma koreksi ejaan *Peter Norvig*. Proses BT menghasilkan tambahan instance sebesar 25% setelah dilakukan filtrasi berbasis kemiripan semantik (*cosine similarity*). Selain augmentasi, peneliti juga menerapkan SMOTE untuk *upsample* kelas jawaban salah sebesar 10%. Studi ini membandingkan dua variasi *embedding multilingual SBERT* dan *IndoSBERT* dan menggunakan Optuna (TPE) untuk optimasi *hyperparameter*. Model terbaik yang dilaporkan merupakan NN dengan tiga *hidden layers*, yang berhasil meningkatkan F1-*score* maksimal pada soal B menjadi 77,2% dan pada soal A menjadi 89,7%, sehingga F1-*score* keseluruhan model mencapai 83,4%, melampaui hasil penelitian berbasis SBERT sebelumnya.

Penelitian terbaru oleh Humaira (2025) berhasil mengembangkan sistem AES Bahasa Indonesia dengan memanfaatkan IndoBERT *embedding* sebagai representasi fitur dan *Feedforward Neural Network* (FNN) sebagai model klasifikasi. Fitur yang diuji meliputi hanya *answer embedding* saja maupun kombinasi dengan *distance feature*, yakni jarak absolut antara *embedding* jawaban siswa dan *embedding* jawaban panduan. Untuk menangani ketidakseimbangan label, peneliti menerapkan SMOTE. Tahapan prapemrosesan yang dilakukan terbatas pada tokenisasi. Setelah proses pelatihan dan evaluasi, model terbaik yang dihasilkan mencatat F1-*score* keseluruhan sebesar 76,7%. Secara rinci, model terbaik untuk soal A (dengan SMOTE) memperoleh F1-*score* 83,5%, sedangkan model terbaik untuk soal B mencapai F1-*score* 69,9%.

2.2. Penilaian Otomatis Esai (*Automatic Essay Scoring*)

Penilaian Otomatis Esai atau *Automatic Essay Scoring* merupakan sebuah sistem komputer yang dirancang untuk mengevaluasi dan memberikan nilai

pada tulisan siswa secara otomatis dengan menganalisis berbagai fitur tertentu (Ramesh *and* Sanampudi, 2022). Implementasi sistem ini bertujuan untuk membantu guru atau asesor dalam menilai esai peserta didik tanpa harus melakukannya secara manual satu per satu, mengingat kapasitas manusia memiliki keterbatasan, terutama dalam menjaga konsistensi penilaian. Tujuan inti dari AES adalah untuk menjawab tantangan dalam penilaian konvensional seperti waktu, biaya, dan objektivitas (Lim *et al.*, 2021). Bahkan, dalam skala penilaian yang besar, sistem AES dianggap mampu menghasilkan penilaian yang lebih stabil dan objektif dibandingkan penilai manusia (Pradani dan Suadaa, 2023).

Sistem AES masih memiliki sejumlah keterbatasan, sistem ini beroperasi dengan menganalisis ciri-ciri kebahasaan, baik dari segi sintaksis (gaya bahasa) maupun semantik (isi tulisan) (Ramesh *and* Sanampudi, 2022). Akibatnya, AES rentan terhadap gangguan dalam teks, seperti kesalahan penulisan, tata bahasa, atau struktur kalimat yang tidak teratur, yang berpotensi memengaruhi akurasi penilaian (Mufiid dkk., 2021). Lebih lanjut, mesin ini pada dasarnya hanya mengenali pola linguistik dalam esai tanpa memiliki pemahaman mendalam tentang makna yang disampaikan.

Walaupun demikian, *Automatic Essay Scoring* (AES) tetap banyak digunakan di dunia pendidikan karena kemampuannya dalam meringankan tugas penilaian manual (Hakiki dan Fatichah, 2025). Sistem ini menawarkan beberapa keunggulan, seperti kemampuan memeriksa jawaban esai dengan lebih teliti daripada manusia, konsistensi yang tinggi selama proses evaluasi, serta objektivitas yang tidak terpengaruh faktor subjektif (Kinanti dan Qoiriah, 2020).

Seiring kemajuan teknologi, riset tentang AES terus berkembang. Awalnya sistem hanya mengandalkan aturan sederhana untuk menilai gaya penulisan, namun kini telah berevolusi menggunakan *Artificial Intelligence* yang lebih baik dalam menganalisis makna. Potensi penerapannya pun semakin meluas,

tidak hanya terbatas pada penilaian akademik skala besar, tetapi juga pada bidang-bidang lain yang memerlukan analisis teks dan pemahaman bahasa.

2.3. *Natural Language Processing* (NLP)

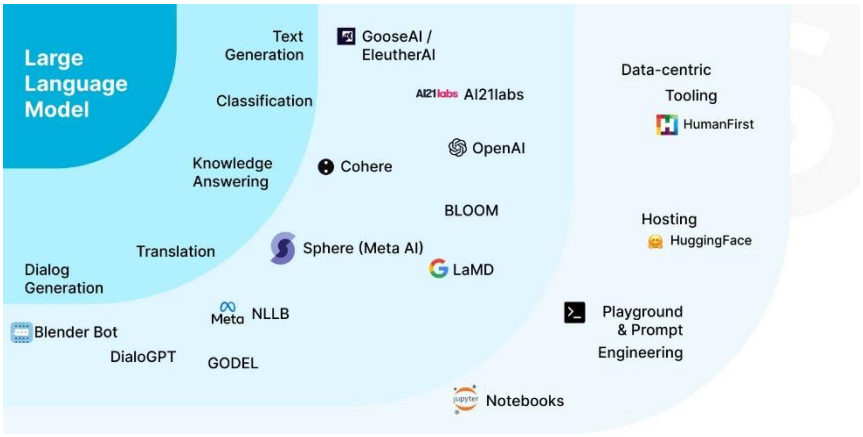
Natural Language Processing (NLP) atau Pemrosesan Bahasa Alami merupakan sub-bidang *Artificial Intelligence* yang berfokus pada penciptaan model komputasi yang meniru kemampuan linguistik (kebahasaan) alami manusia (Abioye *et al.*, 2021). Tujuan utama NLP adalah menjembatani komunikasi antara manusia dan mesin melalui dua pendekatan, yaitu memahami teks (*Natural Language Understanding* / NLU) dan menghasilkan bahasa alami (*Natural Language Generation* / NLG) (Khurana *et al.*, 2023).

Secara umum, NLP diterapkan dalam berbagai fungsi seperti *machine translation*, *information extraction*, *text summarization*, *named entity recognition*, hingga *text categorization*. Dalam proses NLU, mesin melakukan pemrosesan untuk menganalisis dan memahami melalui beberapa tingkat analisis bahasa, mulai dari fonologi (suara), morfologi (struktur kata), sintaksis (tata bahasa), semantik (makna kata), hingga pragmatik (konteks dan maksud). Sebaliknya, proses NLG menghasilkan frasa, kalimat, dan paragraf yang bermakna dari representasi internal (Khurana *et al.*, 2023). Melalui kedua tahapan ini, sistem dapat memahami arti kalimat dan konteks di balik suatu teks.

Seiring dengan perkembangan teknologi, NLP kini banyak mengandalkan pendekatan modern *machine learning* dan *deep learning* untuk mempelajari pola bahasa secara otomatis. Pendekatan modern seperti model *transformer-based* (misalnya BERT hingga LLM) memungkinkan sistem mengenali hubungan semantik antar kata dengan lebih akurat. Dengan demikian, NLP tidak hanya menjadi fondasi bagi berbagai aplikasi berbasis bahasa seperti *chatbot*, tetapi juga berperan penting dalam mendukung interaksi manusia-komputer yang lebih alami dan cerdas.

2.4. *Large Language Models* (LLMs)

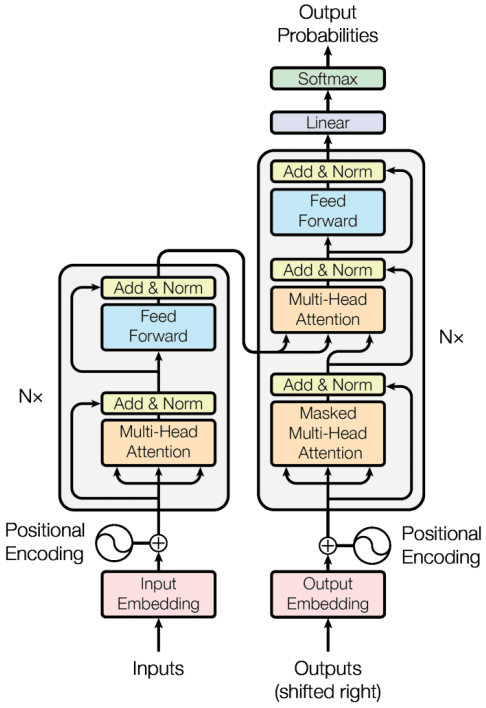
Large Language Models (LLMs) merupakan jenis algoritma kecerdasan buatan yang dirancang untuk memahami dan menghasilkan bahasa alami layaknya manusia. LLM pada dasarnya memiliki kemampuan dalam memproses data teks dalam jumlah besar dan mengenali hubungan semantik antar kata maupun frasa, sehingga dapat memahami konteks dengan sangat baik. Model ini termasuk dalam kategori *Generative AI* karena mampu menghasilkan teks, menerjemahkan bahasa, serta menjawab pertanyaan berdasarkan masukan (*prompt*) yang diberikan (Raiaan *et al.*, 2024; Usman Hadi *et al.*, 2023).



Gambar 1. Ekosistem *Large Language Model* (Vaniukov, 2023).

LLMs beroperasi dengan menerima data teks dari berbagai sumber, yang kemudian diproses melalui tahap tokenisasi (memecah teks menjadi unit diskrit). Setelah diproses, arsitektur LLM *Deep Neural Networks* berbasis *Transformer* dengan miliaran parameter melanjutkan ke proses pelatihan berulang yang melibatkan inisialisasi parameter, perhitungan fungsi kerugian (*loss function*), dan optimasi parameter. LLMs dilatih pada sejumlah besar data teks tak berlabel menggunakan pendekatan pembelajaran mandiri (*self-supervised learning*) (Raiaan *et al.*, 2024).

Terobosan besar dalam pengembangan Large Language Models (LLM) dimulai dengan diperkenalkannya arsitektur *Transformer* melalui publikasi *"Attention is All You Need"* oleh Vaswani *et al.* (2017). Berbeda dari pendekatan sebelumnya yang memproses teks secara berurutan (*sequential processing*), arsitektur *Transformer* (Gambar 2) mampu memproses seluruh urutan kata secara paralel dengan memanfaatkan mekanisme *self-attention*, yaitu kemampuan untuk mengenali dan mempelajari hubungan antar kata di seluruh bagian kalimat tanpa bergantung pada posisi kata.



Gambar 2. Arsitektur *Transformer* (Vaswani *et al.*, 2017).

Mekanisme ini memungkinkan model menangkap dependensi jarak jauh (*long-range dependencies*), seperti hubungan antara subjek dan predikat yang terpisah oleh banyak kata, sehingga pemahaman konteks menjadi lebih akurat. Selain itu, struktur paralelnya membuat proses pelatihan lebih efisien dan *scalable*, karena dapat dijalankan di banyak GPU secara bersamaan, menghasilkan waktu pelatihan yang lebih singkat dan biaya komputasi yang

lebih rendah dibandingkan model berbasis *Recurrent Neural Network*. Berkat keunggulan tersebut, arsitektur *Transformer* menjadi dasar bagi berbagai LLM seperti GPT, BERT, dan Llama, yang menunjukkan performa luar biasa dalam memahami konteks, menghasilkan teks alami, serta menyelesaikan beragam tugas pemrosesan bahasa secara adaptif dan efisien.

Meskipun memiliki performa yang luar biasa, LLM masih menghadapi sejumlah tantangan dalam penerapannya, seperti halusinasi (*hallucination*) yang menghasilkan informasi keliru, serta isu keamanan, privasi, dan serangan *adversarial* yang dapat dimanfaatkan secara negatif. Selain itu, biaya komputasi tinggi dan kebutuhan sumber daya besar menjadi kendala utama dalam pengembangannya. Oleh karena itu, meskipun potensinya sangat besar, pemanfaatan LLM perlu dilakukan secara etis, aman, dan berkelanjutan agar manfaatnya dapat dimaksimalkan tanpa mengabaikan risikonya.

2.4.1. LLaMa 3 *Open-Source* LLM

Open-Source LLM merupakan sebuah model LLM yang dikembangkan secara terbuka dan dapat diakses oleh siapa saja, dengan tujuan untuk mendorong transparansi serta kolaborasi dalam pengembangan kecerdasan buatan. Berbeda dari *Closed-Source Model* seperti GPT-4 yang tidak memperlihatkan proses pengambilan keputusan (Manchanda *et al.*, 2025). *Open-source* LLM memberikan akses penuh terhadap struktur model, metode pelatihan, hingga parameter internal, sehingga peneliti dapat menyesuaikan serta mengoptimalkan model sesuai kebutuhan.

Kemunculan *open-source* LLM dilatarbelakangi oleh keinginan untuk menciptakan sistem AI yang lebih transparan dan mudah digunakan. Salah satu *open-source* LLM paling berpengaruh saat ini adalah *Large Language Model Meta AI* 3 atau Llama 3, sebuah model LLM

yang dikembangkan oleh Meta. Meta merilis dua versi utama, yaitu model yang telah dilatih sebelumnya (*pretrained*) dan model yang telah disesuaikan untuk instruksi (*instruction fine-tuned*), masing-masing tersedia dalam ukuran 8 miliar (8B) dan 70 miliar (70B) parameter. Model ini menggunakan arsitektur *decoder-only transformer* dengan tokenizer 128K yang lebih efisien, dilatih pada lebih dari 15 triliun token multibahasa, serta ditingkatkan menggunakan teknik seperti *Supervised Fine-Tuning* (SFT), *Proximal Policy Optimization* (PPO), dan *Direct Preference Optimization* (DPO) (Meta AI, 2024). Dengan kemampuan pemahaman konteks, penalaran, dan pemrosesan bahasa yang lebih baik dibanding pendahulunya, Llama 3 semakin memperkuat posisi *open-source* LLM sebagai alternatif kuat terhadap model komersil.

2.5. Prapemrosesan Teks (*Text Preprocessing*)

Prapemrosesan teks adalah langkah awal dalam proses *text mining* dan pemrosesan bahasa alami yang bertujuan mempersiapkan data mentah sebelum diproses oleh model NLP. Pada tahap ini, teks yang masih tidak terstruktur, mengandung noise, atau inkonsisten diubah menjadi bentuk yang lebih rapi, seragam, dan siap dianalisis. Proses tersebut membantu model menghasilkan keluaran yang lebih akurat serta mampu merepresentasikan informasi secara lebih tepat.

Large Language Model (LLM) memiliki kemampuan pemahaman konteks teks secara mendalam, sehingga dinilai mampu menangani variasi kata, singkatan, maupun penggunaan bahasa informal, dibandingkan pendekatan model tradisional yang biasanya perlu dibersihkan terlebih dahulu. Hal ini membuat tren cenderung meremehkan atau mengabaikan tahap *preprocessing* teks. Penelitian oleh Siino *et al.* (2024) berjudul "*Is text preprocessing still worth the time? A comparative survey on the influence of popular*

preprocessing methods on Transformers and traditional classifiers" menunjukkan bahwa *preprocessing* tetap dapat memberikan dampak signifikan terhadap kinerja model *pre-train* modern berbasis *Transformer*. Pengaruh tersebut bergantung pada karakteristik dataset serta kombinasi teknik yang diterapkan. Temuan ini menegaskan bahwa pemilihan strategi prapemrosesan yang tepat tetap berperan penting dalam meningkatkan performa model sekaligus meminimalkan potensi bias pada data.

2.5.1. *Data Cleaning*

Data cleaning merupakan langkah pertama dalam proses *preprocessing* yang berfokus pada pembersihan data dari elemen-elemen yang tidak relevan. Dalam konteks persiapan *fine-tuning* sebuah LLM, *data cleaning* tidak secara aktif dilakukan karena dapat memengaruhi pemahaman konteks, sehingga beberapa teknik *cleaning* yang dilakukan mencakup penanganan data duplikat serta penanganan nilai kosong (*null values*) karena dataset yang kosong kemungkinan akan memengaruhi hasil pengujian (Meeradevi *et al.*, 2024).

Dalam konteks LLM, penanganan karakter secara berlebihan dapat secara substansial menurunkan kemampuan model *pre-trained transformer* dalam memahami konteks kalimat secara utuh (Siino *et al.*, 2024). Penelitian oleh Khairani dkk. (2024) menemukan bahwa model berbasis *Transformer* yang tidak melalui tahapan *preprocessing* konvensional seperti *remove stopwords* dan *stemming* menghasilkan kinerja yang lebih baik dibandingkan model yang melalui kedua tahapan tersebut. Secara umum, tahapan ini dilakukan untuk menjaga kualitas dan konsistensi data agar model tidak terganggu oleh *noise* selama proses pelatihan. Data yang bersih menjadi fondasi penting untuk memastikan hasil analisis dan *fine-tuning* berjalan optimal.

2.5.2. Normalisasi Data

Dalam konteks pemrosesan data teks, normalisasi data bertujuan untuk menyamakan bentuk representasi teks dan mengurangi variasi yang tidak diperlukan, seperti penggunaan kata tidak baku, atau bentuk slang. Menurut (Khoerunnisa *et al.*, 2025) normalisasi berperan penting dalam menjaga konsistensi representasi kata dan mencegah perbedaan semantik yang dapat menurunkan akurasi model.

2.5.3. *Misspelling Correction*

Koreksi kesalahan ejaan (*misspelling correction*) adalah langkah penting untuk memastikan kualitas semantik data. Kesalahan ejaan dan tata bahasa dapat dianggap sebagai *noise* yang dapat mengubah semantik kata dan menyesatkan model klasifikasi. Dengan melakukan koreksi terhadap kata yang salah eja dan menyatukan kata-kata yang bermakna serupa, model dapat mengenali pola bahasa dengan lebih tepat, sehingga meningkatkan efektivitas dan akurasi prediksi (Pratap *et al.*, 2025).

Salah satu pendekatan yang dipakai untuk menerapkan koreksi kesalahan ejaan adalah *Peter Norvig Spelling Correction* (Norvig, 2007), sebuah metode sederhana berbasis probabilitas yang mencari kata pengganti paling mungkin benar dengan mempertimbangkan kesamaan bentuk dan frekuensi kemunculan dalam korpus. Teknik ini menghasilkan kandidat koreksi melalui konsep *edit distance*, yakni perbedaan huruf akibat penghapusan, penambahan, penggantian, atau pertukaran posisi. Dari semua kandidat yang dihasilkan, sistem kemudian memilih kata paling umum sebagai hasil koreksi akhir.

2.5.4. Augmentasi Data

Augmentasi Data (*Data Augmentation*) adalah teknik yang digunakan untuk memperbanyak data pelatihan secara sintetis tanpa harus mengumpulkan data baru. Teknik ini menghasilkan teks buatan dengan mengubah teks yang sudah ada. Pendekatan ini sering digunakan dalam berbagai penelitian, terutama untuk mengatasi masalah dataset yang terbatas atau tidak seimbang (*imbalance dataset*). Dengan memperluas distribusi data, teknik augmentasi dapat meningkatkan kinerja model, dan mencegah risiko *overfitting* (Desiani *et al.*, 2023).

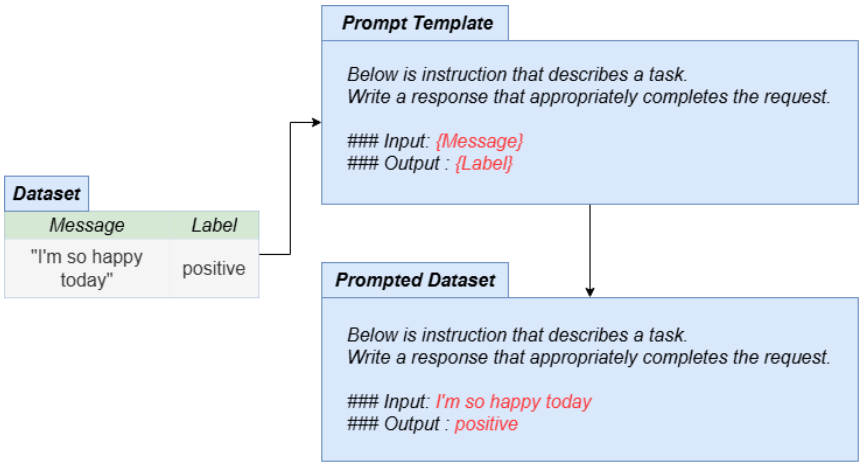
2.5.4.1. *Back Translation Augmentation*

Back Translation merupakan salah satu teknik augmentasi teks yang dilakukan dengan menerjemahkan teks ke bahasa lain lalu mengalihbahasakannya kembali ke bahasa semula, sehingga dapat menghasilkan variasi data teks dengan penggunaan kata yang berbeda (Beddiar *et al.*, 2021). Proses ini tetap mempertahankan makna utama, namun memiliki struktur kalimat atau pilihan kata yang berbeda. Pendekatan ini banyak digunakan dalam NLP, terutama untuk memperluas data pelatihan pada model berbahasa Inggris karena statusnya sebagai bahasa standar internasional. Namun, keterbatasan utamanya adalah ketidakmampuannya menangani idiom dan ekspresi, yang dapat menghasilkan struktur kalimat atau makna yang salah antar bahasa (Desiani *et al.*, 2023).

2.5.5. *Prompt Templating*

Prompt Templating merupakan salah satu komponen penting dalam proses *preprocessing* sebelum melakukan *fine-tuning* pada *Large*

Language Models (LLMs), yang berfungsi untuk mengarahkan model agar menghasilkan keluaran sesuai dengan konteks dan tujuan tertentu. Secara umum, *prompt templating* merujuk pada penggunaan struktur masukan (*input structure*) yang telah ditetapkan sebelumnya dalam bentuk *template* atau pola kalimat tertentu, yang digunakan untuk membentuk hubungan yang konsisten antara instruksi konteks, dan keluaran (*output*) yang diharapkan model (Parthasarathy *et al.*, 2024).



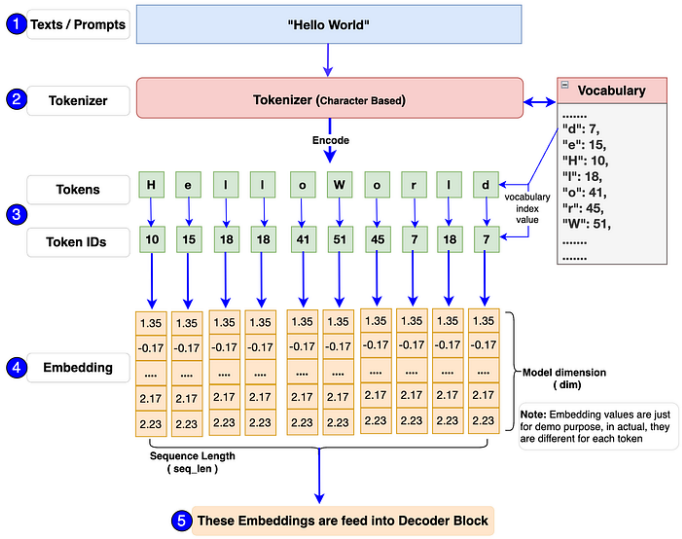
Gambar 3. Proses *Prompt Templating*.

Dalam konteks *fine-tuning*, setiap data pelatihan umumnya disusun dalam pasangan input dan *output* yang mencerminkan perilaku atau respons ideal yang diharapkan dari model. Melalui penggunaan *template*, proses pelatihan menjadi lebih terarah karena model dilatih untuk mengenali pola instruksi tertentu dan memberikan jawaban yang sesuai.

Dari sisi teknis, penerapan *prompt templating* juga bertujuan menjaga konsistensi dan kompatibilitas dataset dengan standar tertentu, misalnya standar format data pada Hugging Face untuk model seperti Llama (Veena *et al.*, 2025). Format ini memastikan bahwa input dan

output memiliki struktur yang seragam, sehingga proses pelatihan dapat dilakukan secara efisien dengan hasil yang stabil. Dalam praktiknya, pembuatan *template* disesuaikan dengan jenis tugas, seperti klasifikasi teks, tanya jawab, atau generasi esai, agar model dapat memahami konteks instruksi dengan lebih baik.

2.6. Tokenisasi

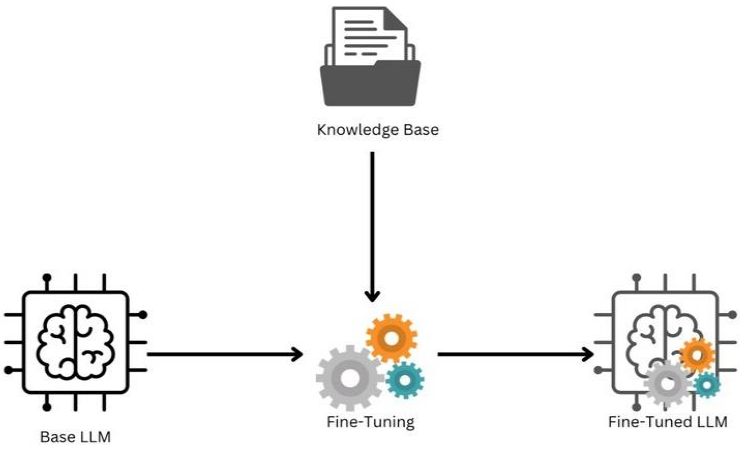


Gambar 4. Alur *Prompt, Tokenizer,* dan *Embedding* (Tamang, 2024).

Secara umum, model arsitektur *Transformer*, termasuk *Large Language Models* (LLM), tidak dapat memproses teks dalam bentuk *string* mentah secara langsung. Oleh karena itu, sebelum memasuki tahap pelatihan, data teks harus melalui proses tokenisasi. Tokenisasi adalah proses pemecahan teks menjadi unit-unit yang lebih kecil yang disebut token, unit ini dapat berupa kata (*word-level*), karakter (*character-level*), maupun subkata (*subword-level*), tergantung pada metode tokenisasi yang digunakan (Tunstall *et al.*, 2022).

Setelah teks dipecah menjadi token, setiap token akan diubah menjadi representasi numerik berupa bilangan bulat yang disebut token ID. Token ID ini kemudian dipetakan ke dalam representasi vektor melalui proses *embedding* sebelum diproses oleh model. Melalui representasi vektor tersebut, model dapat mempelajari hubungan semantik antar token berdasarkan urutannya dalam suatu konteks.

2.7. *Fine-Tuning*



Gambar 5. Ilustrasi proses *Fine-Tuning* LLM.

Fine-Tuning merupakan proses lanjutan dari pelatihan (*Large Language Model*) yang telah melalui tahap *pre-training*. Pada tahap *pre-training*, model dilatih menggunakan data tidak berlabel dalam jumlah besar untuk mempelajari pola, struktur, dan hubungan antar kata dalam bahasa alami. Proses ini berfungsi sebagai inisialisasi bobot agar model memiliki pemahaman dasar terhadap bahasa. Setelah proses *pre-training* selesai, model tersebut belum sepenuhnya optimal untuk tugas tertentu karena masih bersifat umum. Oleh karena itu, diperlukan tahap *fine-tuning* agar model dapat beradaptasi dengan kebutuhan yang lebih spesifik (Awalina dkk., 2022).

Tujuan utama dari *fine-tuning* adalah untuk mengkhususkan model (*specialist*) atau mengadaptasinya pada tugas atau domain spesifik (*specialized use cases*) (Parthasarathy *et al.*, 2024; Pratap *et al.*, 2025).

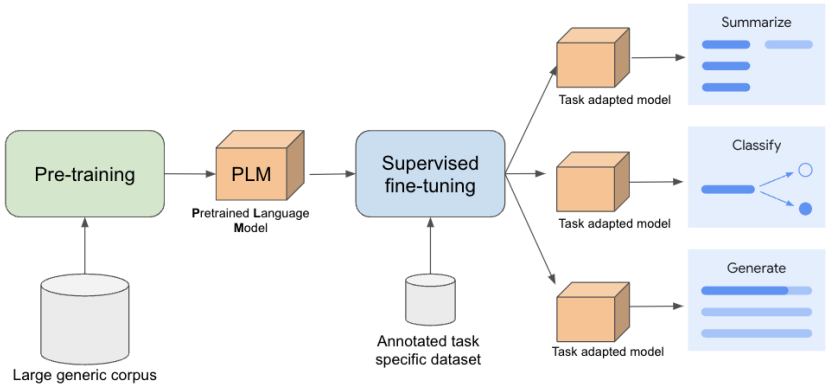
2.7.1. Data *Imbalance Handling*

Data dikatakan tidak seimbang (*imbalance*) terjadi ketika distribusi antar kelas tidak seimbang, di mana jumlah data pada satu kelas jauh lebih banyak dibandingkan kelas lainnya. Kondisi ini dapat menyebabkan model menjadi bias terhadap kelas mayoritas dan berdampak pada penurunan akurasi keseluruhan (Vimalraj *and* R, 2018). Menurut studi (Ali *et al.*, 2019) terdapat tiga pendekatan utama untuk menangani masalah ini, yaitu *data-level approach*, *algorithm-level approach*, dan *cost-sensitive approach*.

Pendekatan *data-level* dilakukan melalui teknik *resampling*, seperti *undersampling* untuk mengurangi jumlah data pada kelas mayoritas, *oversampling* untuk menambah data pada kelas minoritas (misalnya dengan SMOTE), atau kombinasi keduanya. Pendekatan *algorithm-level* dilakukan dengan menyesuaikan algoritma agar tidak bias terhadap kelas mayoritas (Kaope *and* Pristyanto, 2023). Sementara itu, *cost-sensitive approach* memberikan bobot kesalahan yang lebih besar pada kelas minoritas agar model lebih berhati-hati dalam melakukan prediksi. Salah satu implementasi dari pendekatan ini adalah *class-weighted classification*, di mana setiap kelas diberi bobot proporsional terhadap distribusi datanya sehingga model dapat belajar secara lebih seimbang (Brownlee, 2020).

2.7.2. *Supervised Fine-Tuning*

Supervised Fine-Tuning (SFT) merupakan salah satu paradigma dalam proses *fine-tuning* yang menggunakan data berlabel (*labelled data*) untuk melatih model bahasa besar (LLM). Pada metode ini, setiap data masukan (*input*) dikaitkan dengan label atau keluaran yang benar sehingga model dapat belajar hubungan antara keduanya secara langsung. Proses pelatihan dilakukan dengan menyesuaikan parameter model agar mampu memprediksi label yang sesuai berdasarkan data masukan yang diberikan. Pendekatan ini bertujuan untuk meningkatkan kemampuan model dalam menangani tugas-tugas spesifik seperti klasifikasi teks, analisis sentimen, atau *question answering*, di mana hasil yang diharapkan sudah didefinisikan dengan jelas (Anisuzzaman *et al.*, 2025).



Gambar 6. Ilustrasi implementasi *Supervised Fine-Tuning* (Huizenga *and* Hu, 2024).

Meskipun terbukti efektif dalam mengarahkan perilaku model agar sesuai dengan konteks tugas tertentu, pendekatan SFT memiliki tantangan tersendiri. Proses pelatihannya memerlukan dataset berlabel dalam jumlah besar dan berkualitas tinggi, yang sering kali memakan waktu dan biaya besar untuk dikumpulkan. Misalnya, ketika model

ingin disesuaikan untuk tugas klasifikasi teks di ranah bisnis, diperlukan kumpulan data berupa potongan teks dengan label kelas yang sesuai. Oleh karena itu, meskipun SFT memberikan hasil yang presisi dan terarah, efisiensi dan skalabilitasnya sangat bergantung pada ketersediaan dan kualitas data pelatihan yang digunakan (Parthasarathy *et al.*, 2024).

2.7.3. *Hyperparameter*

Hyperparameter merupakan serangkaian parameter yang perlu disesuaikan selama proses *fine-tuning* untuk mengoptimalkan kinerja model bahasa besar (LLM). Dengan pengaturan yang optimal, model mampu menghasilkan generalisasi yang baik ketika diterapkan pada data baru, sehingga performanya tetap stabil dan reliabel (Anisuzzaman *et al.*, 2025). Penyetelan *hyperparameter* yang tepat sangat krusial agar model dapat belajar secara efisien tanpa mengalami *overfitting* (terlalu menyesuaikan diri dengan data pelatihan) maupun *underfitting* (gagal mengenali pola penting pada data).

Penyesuaian *hyperparameter* (*hyperparameter optimizing*) merupakan aspek penting dalam meningkatkan performa model bahasa besar (LLM). Beberapa *hyperparameter* kunci meliputi laju pembelajaran (*learning rate*), ukuran *batch* (*batch size*), *optimizer*, hingga jumlah *epoch*. Untuk mendapatkan kombinasi parameter terbaik, selain dengan menentukan penyetelan parameter secara manual beberapa metode *hyperparameter tuning* yang digunakan, seperti *random search*, *grid search*, dan *Bayesian optimization*, yang membantu mengotomatiskan proses pencarian konfigurasi optimal dan meningkatkan efisiensi pelatihan model.

2.7.3.1. *Learning Rate*

Dalam proses pelatihan model seperti *fine-tuning,* pengaturan *learning rate* memiliki peran penting karena menentukan stabilitas dan kecepatan model dalam mencapai hasil optimal. *Learning rate* menentukan seberapa cepat model beradaptasi terhadap permasalahan. Nilai *learning rate* yang kecil membuat proses pelatihan memerlukan lebih banyak iterasi karena penyesuaian bobot dilakukan secara bertahap, sedangkan *learning rate* yang besar menyebabkan perubahan bobot terjadi lebih cepat (Parthasarathy *et al.*, 2024).

2.7.3.2. *Batch Size*

Batch merupakan sekumpulan data pelatihan yang digunakan untuk memperbarui bobot model selama proses pelatihan. Pelatihan berbasis *batch* dilakukan dengan membagi seluruh dataset pelatihan menjadi beberapa kelompok kecil, lalu memperbarui model setelah setiap *batch* selesai diproses. *Batch size* adalah *hyperparameter* yang menentukan jumlah sampel yang diproses sebelum parameter model diperbarui (Parthasarathy *et al.*, 2024).

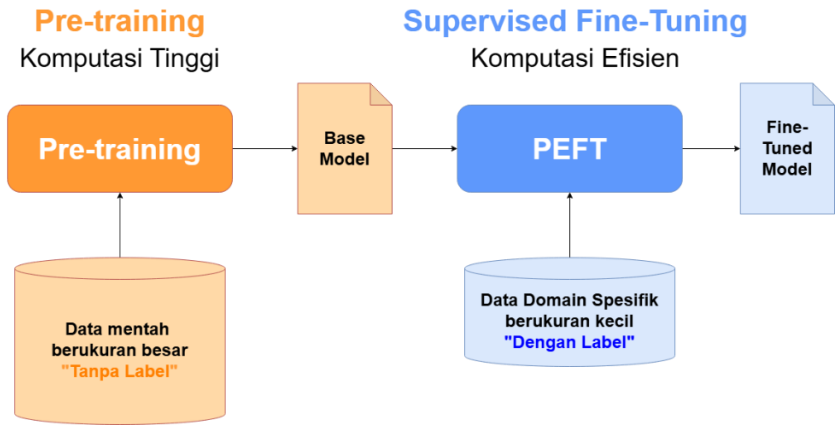
2.7.3.3. *Epoch*

Menunjukkan berapa kali seluruh dataset digunakan selama pelatihan. *Epoch* merujuk pada satu kali putaran penuh pemrosesan seluruh dataset pelatihan. Proses ini mencakup satu kali *forward pass* dan *backward pass* terhadap seluruh data. Dataset dapat diproses sekaligus dalam satu *batch* besar atau dibagi menjadi beberapa *batch* kecil. Satu *epoch* dianggap selesai ketika model telah

memproses semua *batch* dan memperbarui parameternya berdasarkan nilai *loss* yang dihitung (Parthasarathy *et al.*, 2024).

2.7.4. *Parameter-Efficient Fine-Tuning* (PEFT)

Parameter Efficient Fine-Tuning (PEFT) merupakan pendekatan yang memungkinkan model mempelajari tugas baru dengan pembaruan parameter yang minimal. Dalam metode ini, model *pre-trained* hanya disesuaikan melalui pembaruan sebagian kecil parameter tambahan atau terpilih (Liu *et al.*, 2022). Meskipun LLM memiliki miliaran parameter, PEFT memungkinkan pencapaian performa optimal dengan efisiensi tinggi. Pendekatan ini menyeimbangkan antara akurasi dan efisiensi dengan hanya memperbarui sebagian parameter, memanfaatkan pengetahuan dari *base model* (*knowledge distillation*), dan mengoptimalkan struktur model agar tidak ada bagian yang bekerja dua kali (redundansi struktural) (Balne *et al.*, 2024). Hal ini berarti PEFT membuat proses *fine-tuning* menjadi lebih hemat sumber daya tanpa mengorbankan akurasi, karena hanya bagian penting dari model saja yang dilatih, bukan semuanya.



Gambar 7. Alur proses implementasi *Parameter Efficient Fine-Tuning*.

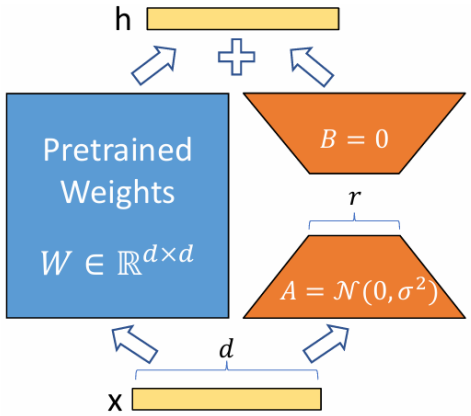
Tujuan utama dari PEFT adalah meningkatkan efisiensi pelatihan dan penggunaan model besar dengan mengurangi biaya komputasi, konsumsi memori, serta kebutuhan penyimpanan model. Karena hanya sebagian kecil parameter yang dilatih, proses *fine-tuning* menjadi lebih ringan dan efisien, bahkan pada perangkat dengan sumber daya terbatas. Dalam banyak kasus, performa PEFT dapat menandingi atau melampaui *full fine-tuning*, menjadikannya solusi ideal untuk pengembangan LLM berskala besar (Hu *et al.*, 2021; Liu *et al.*, 2022).

Namun, PEFT juga memiliki sejumlah keterbatasan. Menurut penelitian (Lialin *et al.*, 2024) untuk model yang relatif kecil, metode PEFT yang menambahkan banyak parameter dapat menyebabkan pelatihan menjadi lebih lambat dibandingkan *full fine-tuning.* Selain itu, PEFT terkadang sensitif terhadap pemilihan *hyperparameter* dan bisa lebih sulit dioptimalkan dibandingkan *fine-tuning* penuh, terutama pada model berukuran kecil. Meski begitu, dengan berbagai variasi metode seperti *LoRA*, *QLoRA, Prefix-Tuning*, *Adapter*, hingga *Hybrid approaches*, PEFT tetap menjadi salah satu inovasi paling penting dalam menjembatani kebutuhan efisiensi dan performa pada era model bahasa besar yang terus berkembang pesat.

2.7.4.1. *Low-Rank Adaptation* (LoRA)

Low-Rank Adaptation (LoRA) merupakan salah satu metode dalam *Parameter-Efficient Fine-Tuning* (PEFT) yang pertama kali diperkenalkan oleh Microsoft melalui penelitian (Hu *et al.*, 2021). Penelitian tersebut menjelaskan bahwa LoRA memungkinkan proses pelatihan pada sebagian *dense layer Neural Network* dilakukan secara tidak langsung, yakni dengan mengoptimalkan matriks dekomposisi berperingkat rendah (*low-rank decomposition*

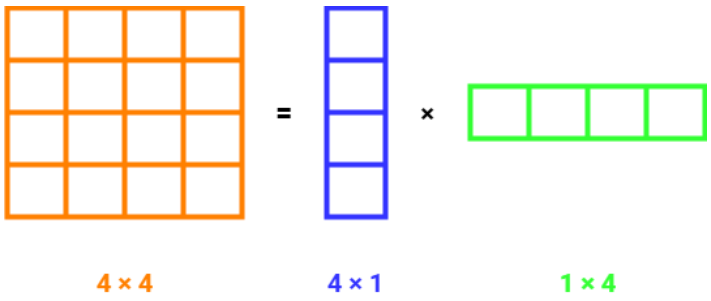
matrices) yang merepresentasikan perubahan bobot selama proses adaptasi, sementara bobot model pra-latih tetap dibekukan dan tidak diperbarui.



Gambar 8. Ilustrasi *Low-Rank Adaptation* (Hu *et al.*, 2021).

Model *deep learning*, termasuk *Large Language Models (LLMs)*, bergantung pada matriks bobot yang menyimpan parameter hasil pembelajaran selama tahap *pre-training*. Pada proses *fine-tuning*, matriks bobot (W) diperbarui secara langsung. Namun, LoRA memperkenalkan pendekatan berbeda dengan merepresentasikan pembaruan bobot (ΔW) sebagai hasil perkalian dua matriks berperingkat rendah (Wa dan Wb) sehingga $\Delta W = Wa \times Wb$. Secara empiris, penelitian menunjukkan bahwa sebagian besar model pra-latih memiliki *dimensi intrinsik* yang rendah, sehingga re-parameterisasi berdimensi rendah tersebut dapat memberikan hasil *fine-tuning* yang setara dengan pembaruan penuh pada seluruh parameter.

Penerapan LoRA dilakukan melalui tiga tahap utama agar proses *fine-tuning* menjadi lebih efisien.



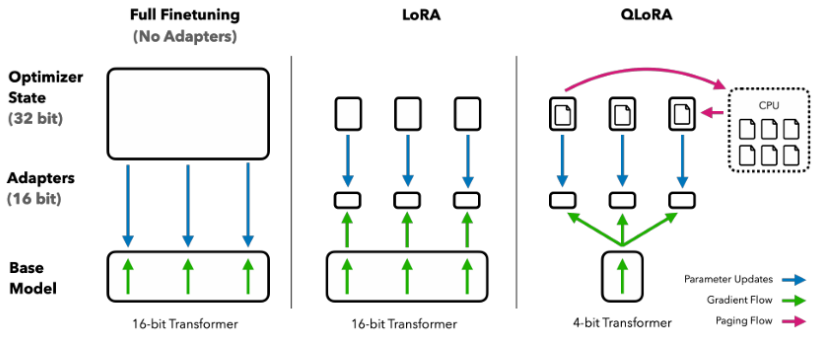
Gambar 9. Ilustrasi proses *Decomposition* matriks pada LoRA.

a. *Decomposition*: Matriks pembaruan bobot ΔW dipecah menjadi dua matriks yang lebih kecil, yaitu A dan B. Nilai r (*rank*) menjadi pengatur seberapa besar ukuran kedua matriks tersebut. Berdasarkan teori *Rank Factorization* dalam aljabar linear, seperti pada gambar Gambar 9, matriks besar bisa diuraikan menjadi perkalian dua matriks kecil dengan ukuran tertentu, selama *rank*-nya minimal satu.
b. *Training*: Hanya matriks A dan B yang melalui proses *fine-tuning*, sementara bobot awal model W tetap *frozen* agar tidak berubah.
c. *Merging*: Setelah *fine-tuning* selesai, hasil pembaruan dari A dan B digabungkan kembali ke dalam matriks W, sehingga model dapat mempertahankan performa awalnya.

Secara keseluruhan, LoRA memiliki beberapa keunggulan utama yang membuatnya efisien dan fleksibel dalam proses *fine-tuning*. Model pra-latih dapat digunakan bersama untuk berbagai tugas hanya dengan mengganti matriks A dan B, sehingga menghemat penyimpanan dan mempermudah perpindahan antartugas. Selain itu, LoRA membuat proses pelatihan lebih ringan karena hanya mengoptimasi dua matriks berukuran kecil tanpa perlu menghitung gradien seluruh parameter, sehingga kebutuhan perangkat keras berkurang hingga tiga kali lipat. Desain linear LoRA juga memungkinkan hasil pelatihan digabungkan dengan bobot asli tanpa menambah waktu inferensi.

2.7.4.2. *Quantized Low-Rank Adaptation* (QLoRA)

QLoRA pertama kali diperkenalkan melalui artikel (Dettmers *et al.*, 2023). QLoRA merupakan salah satu metode *Parameter-Efficient Fine-Tuning (PEFT)* yang dikembangkan untuk mengatasi keterbatasan memori dalam pelatihan model berukuran besar. Metode ini menggabungkan efisiensi *quantization* dengan pendekatan *low-rank adaptation* dari LoRA. Secara prinsip, QLoRA membekukan model pra-latih dalam bentuk bobot terkuantisasi dengan presisi *4-bit*, kemudian melakukan propagasi gradien (*backpropagation*) melalui bobot terkuantisasi tersebut menuju *Low-Rank Adapters (LoRA)* yang dapat dilatih.



Gambar 10. Ilustrasi *Quantized Low-Rank Adaptation* (Dettmers *et al.*, 2023).

QLoRA bekerja melalui tiga tahap utama. Pertama, *Quantization*, yaitu mengubah bobot model menjadi representasi *4-bit* menggunakan teknik NF4 untuk menghemat memori tanpa kehilangan informasi penting. Kedua, *LoRA Fine-Tuning*, di mana dilakukan *re-rank matriks* seperti pada LoRA, sementara bobot asli tetap dalam kondisi terkuantisasi dan tidak diubah. Ketiga, *Preserved Performance*, berbeda dengan LoRA yang melakukan *merging* pada hasil pelatihan ke bobot utama, QLoRA

mempertahankan *adaptor* LoRA secara terpisah di atas model yang telah dikuantisasi. Pendekatan ini tetap menjaga akurasi dan stabilitas model, meskipun penggunaan memori dan sumber daya jauh lebih efisien.

2.7.5. Hugging Face

Hugging Face merupakan perusahaan sekaligus komunitas *open-source* yang berfokus pada pengembangan alat, model pembelajaran mesin, dan platform untuk bekerja dengan *artificial intelligence*, khususnya pada bidang data science, *machine learning*, serta *natural language processing* (NLP) (Stryker, 2025). Komunitas pada *platform* ini secara rutin berkontribusi dalam bentuk model AI baru, dataset, tutorial, hingga riset. Sebagai *platform* sekaligus pustaka *machine learning*, Hugging Face menyediakan beragam *pre-trained model* yang dapat dimanfaatkan untuk berbagai tugas seperti *conversational* AI, analisis sentimen, klasifikasi teks, maupun *computer vision* (Coursera, 2025). Dengan pendekatan *open-source*, Hugging Face mempercepat adopsi teknologi AI dan memperluas kolaborasi dalam pengembangan solusi berbasis pembelajaran mesin.

2.7.6. Unsloth

Unsloth merupakan kerangka kerja (*Framework*) *Open-Source* yang dikembangkan untuk mempermudah proses *fine-tuning Large Language Model* (LLM) serta penerapan *Reinforcement Learning* (RL). Tujuan utama Unsloth adalah menghadirkan *Artificial Intelligence Resource* yang efisien, akurat, dan mudah diakses. Unsloth diklaim mampu melatih model dua kali lebih cepat dengan konsumsi VRAM lebih hemat hingga 70%, serta mempertahankan akurasi penuh. Selain efisien dan akurat, Unsloth memiliki dukungan

luas untuk berbagai jenis model dan mode pelatihan, mulai dari *full fine-tuning*, *pre-training*, hingga pelatihan dalam presisi *4-bit*, 8-bit, dan 16-bit (Unsloth, 2025).

Unsloth memungkinkan pengguna untuk melatih, mengevaluasi, menyimpan, hingga mengintegrasikan berbagai model seperti GPT-OSS, Llama, DeepSeek dan lain-lain. Proses kerjanya mencakup tahap *loading*, *quantization*, pelatihan, hingga ekspor model ke mesin inferensi seperti Ollama, *Llama*.cpp, dan vLLM. *Framework* ini juga dirancang untuk kompatibel dengan berbagai sistem operasi dan perangkat keras, seperti Linux, Windows, Colab, serta GPU dari NVIDIA, AMD, dan Intel. Selain itu, Unsloth juga telah terintegrasi dengan Hugging Face *Transformers* serta menawarkan berbagai optimasi untuk efisiensi pelatihan model (Mansha, 2025).

2.8. *Confusion Matrix*

Confusion Matrix merupakan metode yang digunakan untuk memvisualisasikan serta merangkum hasil evaluasi kinerja dari suatu algoritma klasifikasi. Secara umum, *confusion matrix* adalah tabel statistik yang digunakan untuk menganalisis dan merangkum hasil prediksi model klasifikasi. Bentuk paling dasar dari *confusion matrix* adalah versi *binary*, yang melibatkan dua kelas, seperti *Yes/No*, *Positive/Negative*, atau *Spam/Not Spam*, matriks tersebut biasanya direpresentasikan dalam ukuran 2×2 (AlShammari, 2024), bentuk tersebut bisa dilihat pada Tabel 2.

Dalam *Confusion Matrix*, terdapat empat komponen utama yang merepresentasikan hasil prediksi model klasifikasi, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai TP menunjukkan jumlah data positif yang berhasil diprediksi dengan benar sebagai positif, sedangkan TN menunjukkan jumlah data negatif yang diprediksi dengan benar sebagai negatif. Sebaliknya, FP merupakan jumlah

data negatif yang salah diklasifikasikan sebagai positif, dan FN adalah jumlah data positif yang salah diprediksi sebagai negatif (Kulkarni *et al.*, 2020).

Tabel 2. *Confusion Matrix*

Confusion Matrix		***Predicted***	
		Positive	***Negative***
Actual	*Positive*	TP	FN
	Negative	FP	TN

Confusion Matrix pada dasarnya menampilkan perbandingan antara nilai prediksi model dengan nilai aktual untuk memberikan gambaran mengenai performa model klasifikasi. Dari matriks ini, dapat dihitung berbagai metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan F1-*score*.

2.8.1. Akurasi

Akurasi (*Accuracy*) merupakan ukuran yang menunjukkan seberapa sering model melakukan prediksi dengan benar. Nilai ini diperoleh dengan membandingkan jumlah prediksi yang benar terhadap keseluruhan prediksi yang dilakukan (AlShammari, 2024). Persamaan akurasi dapat dilihat pada persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2.8.2. Presisi

Presisi (*Precision*) menggambarkan tingkat ketepatan model dalam mengidentifikasi kelas positif. Nilai ini dihitung berdasarkan perbandingan antara jumlah prediksi positif yang benar dengan

seluruh prediksi positif yang dihasilkan model (AlShammari, 2024). Persamaan presisi dapat dilihat pada persamaan (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

2.8.3. *Recall*

Recall, atau yang sering disebut *sensitivity* atau *true positive rate*, menunjukkan kemampuan model dalam menemukan seluruh data positif yang sebenarnya. Nilai ini dihitung dari jumlah prediksi positif yang benar dibandingkan dengan jumlah seluruh data positif aktual (AlShammari, 2024). Persamaan *recall* dapat dilihat pada persamaan (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

2.8.4. F1-*score*

F1-*Score* merupakan rata-rata harmonik antara nilai presisi dan *recall* (AlShammari, 2024). F1-*score* memperhatikan kemampuan model dalam menangani kelas data yang tidak seimbang (Rajagede, 2021). Persamaan F1-*score* dapat dilihat pada persamaan (4).

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

III. METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penjelasan mengenai tempat dan periode pelaksanaan penelitian akan dipaparkan sebagai berikut.

3.1.1. Tempat Penelitian

Penelitian ini dilaksanakan di lingkungan Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, yang berlokasi di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Provinsi Lampung.

3.1.2. Waktu Penelitian

Tabel 3. Rencana penelitian

Kegiatan	2025			2026
	Okt	Nov	Des	Jan
Pengumpulan Dataset				
Preprocessing				
Model Preparation				
Supervised Fine-Tuning				
Evaluasi Model				
Penulisan Laporan				

Pelaksanaan penelitian ini berlangsung pada semester ganjil tahun ajaran 2025/2026, dimulai sejak Oktober 2025 hingga Januari 2026. Tahapan pelaksanaan penelitian secara rinci disajikan pada Tabel 3 yang memuat alur kegiatan dari awal hingga akhir penelitian.

3.2. Perangkat Penelitian

Uraian lengkap mengenai perangkat penelitian, baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) yang digunakan, disajikan pada bagian berikut.

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini berupa perangkat laptop dengan spesifikasi dan detail dijelaskan pada Tabel 4. Perangkat ini digunakan untuk menjalankan proses komputasi, eksperimen, hingga evaluasi model.

Tabel 4. Spesifikasi perangkat keras

Spesifikasi	Detail
Merek	HP
Tipe	HP Laptop 14-dk1025wm
Processor	AMD Ryzen™ 3 3300U
Graphics	AMD Radeon Vega 6
RAM	8GB DDR4
Penyimpanan	SSD M.2 2280 256GB

3.2.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam proses penelitian ini adalah sebagai berikut.

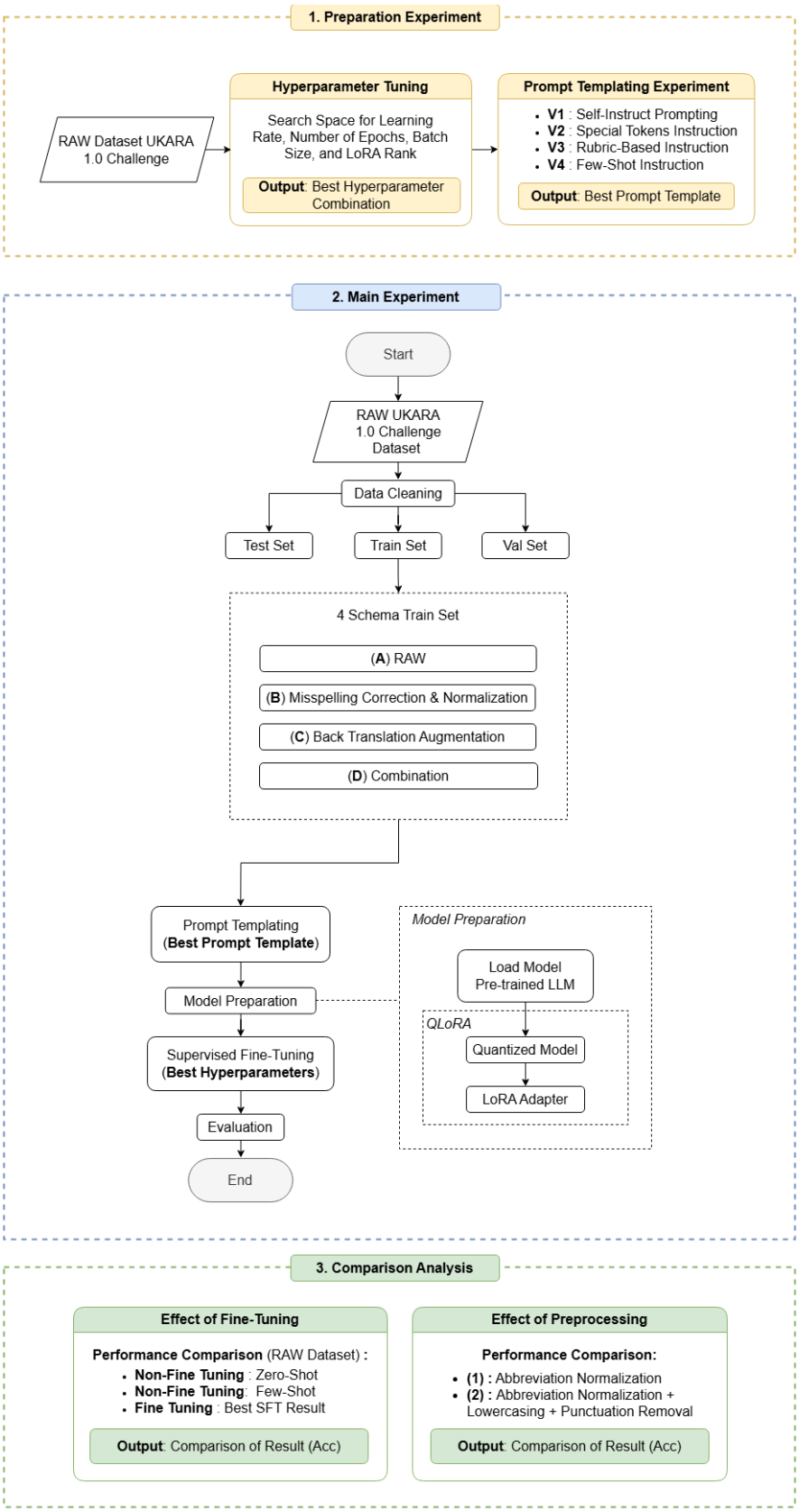
a. Sistem Operasi Windows 11 Home *64-bit*
b. Google Colab *Platform* dengan Python 3.12 serta library Unsloth dan Hugging Face sebagai lingkungan eksperimen utama untuk proses *fine-tuning* model
c. Visual Studio Code (Versi 1.105) sebagai *text editor* dalam proses pengembangan kode dan model
d. Google Drive, dimanfaatkan sebagai media penyimpanan berbagai keperluan penelitian seperti dataset, kode program, hingga berkas pendukung penelitian lainnya
e. Draw.io, *platform* yang digunakan sebagai alat bantu visualisasi diagram alur penelitian dan arsitektur sistem
f. Web Browser, untuk mendukung akses ke layanan Google Colab, Drive, hingga Draw.io

3.3. Tahap Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang tersusun secara sistematis untuk mencapai tujuan yang telah ditetapkan. Alur dari tahapan penelitian tersebut diilustrasikan pada Gambar 11, yang menggambarkan proses mulai dari pengumpulan dataset hingga tahap evaluasi model.

3.3.1. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari kegiatan UKARA 1.0 *Challenge*, UKARA sendiri merupakan sebuah proyek penelitian oleh tim *Natural Language Processing* (NLP) Universitas Gadjah Mada (UGM) pada tahun 2018 (Herwanto dkk., 2018). Riset tersebut dilanjutkan dengan diselenggarakannya UKARA *Challenge* 1.0 pada tahun 2019, bekerja sama dengan Pusat Penilaian Pendidikan (PUSPENDIK), Kementerian Pendidikan dan Kebudayaan Indonesia, sekaligus publikasi dataset yang dirancang untuk mendukung pengembangan sistem *Automatic Essay Scoring* berbahasa Indonesia.



Gambar 11. Diagram alur penelitian.

Dataset UKARA terdiri atas jawaban esai siswa yang dinilai secara manual oleh tim ahli dengan dua label utama, yaitu 1 (benar) dan 0 (salah). Penilaian dilakukan berdasarkan kesesuaian isi jawaban terhadap kunci konsep yang telah ditetapkan. Dataset dibagi menjadi dua sub-permasalahan, yakni *Problem* A dan *Problem* B, yang masing-masing memuat satu pertanyaan dengan kumpulan respons siswa. Jumlah data pada *Problem* A adalah 1.338 sampel, sedangkan *Problem* B memiliki 1.523 sampel, sehingga keseluruhan dataset berjumlah 2.861 sampel. Sampel data untuk *Problem* A bisa dilihat di Tabel 5 dan sampel data untuk *Problem* B bisa dilihat di Tabel 6.

Tabel 5. Sampel data UKARA *Problem* A

RESPONSE	LABEL
intetraksi/beradaptasi terhadap lingkungan yang baru	1
seperti jatuhnya meteor tsunami gempa bumi,	0

Tabel 6. Sampel data UKARA *Problem* B

RESPONSE	LABEL
Karena orang berpikir bahwa jika disumbangkan akan membuat produksi pakaian menjadi lebih beretika	1
kerena harganya mahal .	0

3.3.2. *Data Preprocessing*

Tahap *preprocessing* merupakan langkah awal sebelum dataset digunakan dalam proses *fine-tuning* model LLM. Proses ini mencakup beberapa tahapan penting, antara lain *data cleaning*, *misspelling correction*, normalisasi teks, serta augmentasi melalui *back translation*.

3.3.2.1. Data *Cleaning*

Dataset yang telah disiapkan terlebih dahulu akan melalui proses data *cleaning*, sebelum kembali dibagi menjadi data latih, validasi, dan uji. Sedikit berbeda dari teknik pembersihan data konvensional, pada penelitian ini proses *cleaning* dilakukan secara selektif agar tidak menghilangkan konteks linguistik yang diperlukan oleh LLM. Tahapan yang diterapkan mencakup penghapusan duplikasi data, penanganan kolom duplikat, serta penanganan nilai hilang (*missing values*) hingga penghapusan spasi berlebih, seperti yang dapat dilihat pada Tabel 7.

Tabel 7. Contoh penerapan teks *Cleaning*

Teks Asli	"*masyarakat cendrung berpikir kritis*"
Teks *Cleaning*	"*masyarakat cendrung berpikir kritis*"

Namun demikian, beberapa teknik umum seperti *stemming* atau *lemmatization* tidak digunakan karena berpotensi mengurangi kekayaan konteks (*semantic richness*) yang dibutuhkan model selama proses pelatihan.

3.3.2.2. *Misspelling Correction*

Teks yang telah melalui tahap *cleaning* selanjutnya dilakukan proses *misspelling correction* untuk memperbaiki kesalahan ejaan pada data latih. Proses ini dilakukan dengan memuat korpus KBBI sebagai acuan kata baku, kemudian memperbaiki kata yang tidak sesuai. Peneliti menggunakan algoritma koreksi ejaan *Peter Norvig*, berdasarkan pertimbangan penggunaan pada penelitian sebelumnya oleh Tanaka *et al.* (2024) algoritma ini merupakan pendekatan berbasis probabilitas dalam mencari kata pengganti yang paling

mendekati benar berdasarkan kesamaan bentuk dan frekuensi kemunculan dalam korpus KBBI yang dipakai. Contoh penerapannya dapat dilihat pada Tabel 8.

Tabel 8. Contoh penerapan *Misspelling Correction*

Teks Asli	*"pemeritah", "kcing", "indonesa"*
Misspelling Correction	*"pemerintah", "kucing", "indonesia"*

3.3.2.3. Normalisasi Teks

Tahap normalisasi bertujuan untuk menstandarkan bentuk kata tidak baku seperti kata singkatan (*Abbreviations*), bahasa gaul, atau kata slang menjadi bentuk baku sesuai kaidah bahasa Indonesia. Proses ini dilakukan dengan menggunakan kamus/*corpus* kata-kata non-formal (*slang words*) bahasa Indonesia, yang berisi pasangan kata tidak baku dan bentuk normalnya.

Tabel 9. Contoh penerapan normalisasi teks

Teks Asli	*"gak ada bukti klo berita itu bener"*
Normalisasi Teks	*"tidak ada bukti kalau berita itu benar"*

Seperti yang dapat dilihat pada Tabel 9, setiap kata dalam teks diperiksa dan diganti sesuai padanan baku pada korpus tersebut agar model dapat memahami konteks secara lebih konsisten dan semantik tetap terjaga.

3.3.2.4. Augmentasi Data (*Back Translation Augmentation*)

Penelitian ini akan menerapkan proses augmentasi data menggunakan teknik *back translation* untuk memperkaya variasi

data latih. Pendekatan back-translation dipilih karena terbukti masih menjadi metode augmentasi yang unggul dalam menjaga kesamaan makna (*semantic fidelity*) dan mendongkrak F1-*score* dibandingkan metode berbasis LLM generatif (Radliński *et al.*, 2025). Proses ini dilakukan dengan menerjemahkan teks berbahasa Indonesia ke dalam bahasa Inggris, kemudian menerjemahkannya kembali ke bahasa Indonesia. Sebagai contoh penerapan teknik ini dapat dilihat pada Tabel 10.

Sebelum data sintetik hasil augmentasi disimpan, dilakukan tahap seleksi menggunakan perhitungan *cosine similarity* untuk menilai tingkat kemiripan antara teks asli dan hasil *back translation*. Representasi vektor teks akan diperoleh menggunakan Indo-SBERT (*Indonesian Sentence-BERT*), sebagaimana pendekatan yang mengombinasikan SBERT dan *cosine similarity* dalam penelitian Fathuddin dkk. (2025). Pendekatan ini juga mengacu pada penelitian sebelumnya oleh Tanaka (2024) dan (Corbeil & Ghavidel, 2021) yang mana hasil augmentasi dievaluasi kelayakannya menggunakan *threshold* atau batas nilai tertentu, sampel yang gagal memenuhi dalam batas kemiripan tersebut akan dibuang.

Tabel 10. Contoh penerapan *Back Translation*

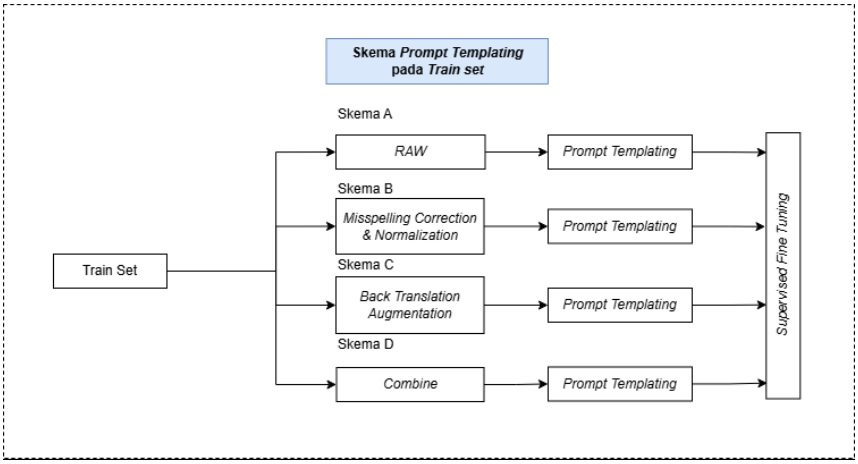
Teks Asli (id)	"Berita itu belum dapat dipastikan."
Terjemahan (en)	"*The news has not been confirmed.*"
Hasil Terjemahan Kembali (id)	"Berita tersebut belum dikonfirmasi"

Cosine similarity sendiri digunakan untuk mengukur kedekatan makna antara dua teks dalam bentuk representasi vektor dengan rentang nilai 0 hingga 1 (Sanjaya dkk., 2023). Hanya pasangan teks dengan tingkat kemiripan yang masih relevan secara semantik

namun tidak identik sepenuhnya yang dipertahankan sebagai data hasil augmentasi.

3.3.2.5. *Prompt Templating*

Selanjutnya setiap skema data yang akan dilatih, terlebih dahulu diubah ke dalam bentuk *prompt,* seperti yang dapat dilihat pada Gambar 12. Berbeda dengan pelatihan *machine learning* konvensional, tahap ini bersifat khusus karena model LLM telah dilatih menggunakan pendekatan berbasis *instruction*. Tahap ini bertujuan untuk mengubah dataset ke dalam format *prompt* yang sesuai dengan gaya pelatihan model LLM *instruction-based*.



Gambar 12. *Prompt Templating* pada tiap skema dataset.

Setiap sampel data diubah menjadi pasangan instruksi (*instruction*), konteks (*input*), dan jawaban (*response*), sehingga model dapat belajar merespons perintah secara natural. *Template* yang digunakan disusun dalam format yang direkomendasikan oleh *library* Unsloth, dan diterapkan pada seluruh data latih serta data validasi sebelum proses *fine-tuning* dimulai.

Penelitian ini menerapkan beberapa variasi *prompt templating*, mulai dari struktur *prompt* sederhana yang mengacu pada format *self-instruct* (Wang dkk., 2023), hingga variasi yang lebih kompleks lainnya dengan memanfaatkan struktur yang lebih terarah, seperti penggunaan *special tokens* dari Llama 3 (Meta AI, 2024). Berbagai variasi tersebut digunakan untuk mengevaluasi konfigurasi *prompt* yang paling efektif dalam meningkatkan kinerja model.

3.3.3. *Model Preparation*

Sebelum proses pelatihan dilakukan, terdapat beberapa tahapan persiapan model yang perlu diselesaikan terlebih dahulu. Model utama yang digunakan pada penelitian ini adalah *Llama 3.1 8 Billion Parameters-Instruct* versi Unsloth (*Finetune Llama 3.1 with Unsloth*, 2024), sebuah model LLM *open-source* yang tersedia melalui platform Hugging Face (Unsloth, 2024). Setelah model berhasil dimuat, peneliti menerapkan pendekatan *Parameter-Efficient Fine-Tuning* (PEFT) dengan metode QLoRA untuk mengefisienkan proses pelatihan tanpa perlu memperbarui seluruh parameter model.

3.3.3.1. *Quantized Low-Rank Adaptation* (QLoRA)

Persiapan model diawali dengan melakukan konfigurasi dalam penerapan pendekatan PEFT dengan metode QLoRA. Pertama model terlebih dahulu melalui proses *quantization* pada model menggunakan pustaka Unsloth. Proses ini mengubah representasi parameter model (dari 32/16-*bit floating point* menjadi format *4-bit Integer*) sehingga ukuran model menjadi lebih ringan dan kebutuhan memori dapat ditekan secara signifikan.

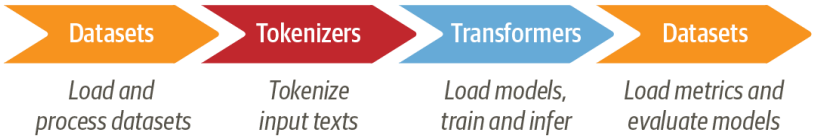
Sebelum model digunakan untuk *fine-tuning*, konfigurasi LoRA *Adapter* juga diatur agar proses pembelajaran berlangsung lebih

optimal. Adapun parameter yang digunakan dalam *Adapter* LoRA dapat dilihat pada Tabel 11.

Tabel 11. *Hyperparameter* pada LoRA *Adapter*

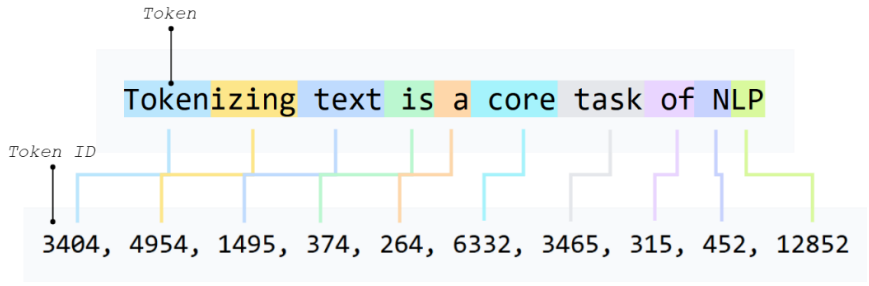
Nama Parameter	Nilai
LoRA *Rank* (*r*)	16, 32
LoRA *Alpha* (*lora_alpha*)	2 * *r*

3.3.4. Tokenisasi



Gambar 13. *Alur pelatihan model Transformer* (Tunstall *et al.*, 2022).

Dataset yang telah diformat ke dalam *prompt template* selanjutnya melalui proses tokenisasi. *Tokenizer* yang digunakan merupakan tokenizer bawaan dari model Llama 3.1, sehingga konsisten dengan proses *pre-training* model tersebut. Llama menggunakan SentencePiece, yaitu metode tokenisasi yang dikembangkan oleh tim Google (Kudo *and* Richardson, 2018). SentencePiece tidak bergantung pada aturan linguistik tertentu karena memperlakukan teks sebagai rangkaian karakter utuh.



Gambar 14. Ilustrasi proses tokenisasi.

Di dalam implementasinya, SentencePiece mendukung algoritma pemotongan berbasis *subword*, seperti *Byte-Pair Encoding* (BPE) dan *Unigram Language Model*. Pendekatan *subword* memungkinkan teks dipecah menjadi unit-unit sub-kata, ilustrasi proses ini dapat dilihat pada Gambar 14. Pada penelitian ini, tokenisasi dilakukan secara otomatis selama proses *fine-tuning* menggunakan *tokenizer* bawaan model. Panjang maksimum sekuens ditetapkan menyesuaikan dengan panjang hasil *prompt template* pada dataset yang digunakan.

3.3.5. *Supervised Fine-Tuning*

Proses *supervised fine-tuning* dilakukan untuk menyesuaikan kemampuan model agar mampu mempelajari pola klasifikasi dari dataset yang telah dipersiapkan sebelumnya. Pada tahap ini, model dilatih menggunakan data berlabel dengan tujuan meningkatkan kemampuan prediksi sesuai dengan instruksi yang diberikan dalam *prompt*.

3.3.5.1. Data *Imbalance Handling*

Salah satu tantangan dalam dataset UKARA 1.0 *Challenge* adalah distribusi label yang tidak seimbang (*imbalanced data*). Kondisi ini berpotensi menyebabkan model cenderung bias terhadap kelas

dengan jumlah data lebih banyak. Untuk mengatasi hal tersebut, penelitian ini menerapkan metode *imbalance handling* pada tingkat *model-level*, yaitu dengan memperbarui fungsi kerugian (*update loss function*). Metode ini memberikan bobot pelatihan yang lebih besar pada kelas dengan jumlah data lebih sedikit, sehingga model dapat belajar secara lebih seimbang dan menghasilkan performa klasifikasi yang lebih adil antar kelas.

3.3.5.2. Optimalisasi *Hyperparameter*

Tabel 12. *Hyperparameter Supervised Fine-Tuning* model

Nama Parameter	Nilai
Learning Rate	5*e*-5, 1*e*-4, 2*e*-4
Number of Epochs	2, 3, 4
Batch Size	8, 16

Dalam penelitian ini, dilakukan proses optimalisasi terhadap beberapa kombinasi *hyperparameter* guna memperoleh hasil kinerja terbaik dari model. Kombinasi *hyperparameter* pada proses *fine-tuning* dapat dilihat pada Tabel 12.

3.3.6. Evaluasi Model

Tahap terakhir yaitu evaluasi kinerja model, evaluasi ini dilakukan menggunakan metode *confusion matrix* dan *classification report* yang mencakup nilai *accuracy*, *precision*, *recall*, dan F1-*score*. Selain berfokus pada metrik akurasi, karena tugas ini termasuk kedalam lingkup *binary classification*, penelitian ini juga berfokus pada performa F1-score kelas positif sebagai standar evaluasi. Hasil ini ditetapkan untuk menilai kinerja model setelah proses *Supervised fine-tuning* serta membandingkannya dengan penelitian-penelitian pada dataset UKARA 1.0 *Challenge* sebelumnya.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Rangkaian penelitian yang telah dilakukan menghasilkan beberapa simpulan sebagai berikut:

1. Penelitian ini berhasil menerapkan *Supervised Fine-Tuning* (SFT) pada *open-source* LLM Meta Llama 3.1 8B *Instruct* untuk tugas penilaian esai otomatis berbahasa Indonesia menggunakan dataset UKARA 1.0 *Challenge*. Penerapan PEFT melalui QLoRA dengan kuantisasi *4-bit* memungkinkan proses pelatihan dilakukan secara efisien pada lingkungan komputasi terbatas tanpa menurunkan kemampuan model secara signifikan.
2. Hasil evaluasi menunjukkan bahwa *fine-tuning* secara signifikan meningkatkan kinerja model dibandingkan pendekatan tanpa *fine-tuning*. Model hasil SFT mencapai akurasi 89,93% pada *Problem* A dan 72,13% pada *Problem* B, dengan F1-*score* terbaik pada kelas positif masing-masing sebesar 93,16% dan 75,68%. Proses pelatihan berjalan stabil dan tidak menunjukkan indikasi *overfitting*.
3. Model yang diusulkan menunjukkan kinerja kompetitif dibandingkan penelitian terdahulu pada dataset yang sama. Model berhasil mencapai F1-*score* tertinggi pada *Problem* A dan performa yang kompetitif pada *Problem* B, dengan rata-rata F1-*score* gabungan sebesar 84,42%, sehingga menempatkannya sebagai salah satu pendekatan dengan performa tinggi dalam penelitian pada dataset UKARA 1.0 *Challenge*.

4. Analisis lanjutan menunjukkan bahwa penggunaan data mentah (*raw* dataset) memberikan kinerja terbaik dibandingkan *preprocessing* dan augmentasi tambahan. Selain itu, *prompt* yang terstruktur secara kontekstual, optimalisasi hyperparameter, serta penerapan *weighted los*s berperan penting dalam meningkatkan performa model dan mengurangi bias akibat ketidakseimbangan data.

5.2. Saran

Saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Mengeksplorasi teknik augmentasi data berbasis LLM (*Generative Augmentation*) untuk memperkaya variasi bahasa dan konteks esai tanpa menghilangkan makna semantik.
2. Fokus penelitian pada peningkatan kinerja untuk *Problem* B, mengingat karakteristik soal yang lebih subjektif dan menuntut pemahaman konteks serta penalaran yang lebih kompleks.
3. Pengembangan model tidak hanya diarahkan pada tugas klasifikasi, tetapi juga pada kemampuan model untuk menghasilkan umpan balik atau alasan penilaian (*explainable feedback*). Untuk mendukung hal tersebut, dataset dapat dikembangkan atau dianotasi ulang dengan menambahkan kolom alasan penilaian.
4. Apabila tersedia sumber daya komputasi yang lebih besar, lakukan perbandingan pendekatan QLoRA dengan *Full Fine-Tuning*, untuk menganalisis perbedaan kinerja, efisiensi, dan kebutuhan komputasi secara lebih komprehensif.
5. Melakukan perbandingan dengan model LLM *open-source* lainnya, seperti Gemma, Qwen, dan Mistral atau dengan model LLM *closed-source* melalui API seperti GPT atau Gemini, untuk memperoleh gambaran performa yang lebih luas.

DAFTAR PUSTAKA

Abioye, S. O., Oyedele, L. O., Akanbi, L., Ajayi, A., Delgado, J. M. D., Bilal, M., Akinade, O. O., & Ahmed, A. (2021). Artificial intelligence in the construction industry: A review of present status, opportunities *and* future challenges. *Journal of Building Engineering*, *44*, 103299. https://doi.org/10.1016/j.jobe.2021.103299

Ali, H., Salleh, M. N. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: a review. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(3), 1560–1571. https://doi.org/10.11591/ijeecs.v14.i3.pp1560-1571

AlShammari, A. F. (2024). Implementation of Model Evaluation using Confusion Matrix in Python. *International Journal of Computer Applications*, *186*(50), 975–8887. https://doi.org/10.5120/ijca2024924236

Anisuzzaman, D. M., Malins, J. G., Friedman, P. A., & Attia, Z. I. (2025). Fine-Tuning Large Language Models for Specialized Use Cases. *Mayo Clinic Proceedings: Digital Health*, *3*(1), 100184. https://doi.org/10.1016/j.mcpdig.2024.11.005

Awalina, A., Bachtiar, F. A., & Utaminingrum, F. (2022). Perbandingan pretrained model transformer pada deteksi ulasan palsu. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *9*(3), 597–604. https://doi.org/10.25126/jtiik.202295696

Awidi, I. T. (2024). Comparing expert tutor evaluation of reflective essays with marking by generative artificial intelligence (AI) tool. *Computers and Education: Artificial Intelligence*, *6*, 100226. https://doi.org/10.1016/j.caeai.2024.100226

Balne, C. C. S., Bhaduri, S., Roy, T., Jain, V., & Chadha, A. (2024). Parameter Efficient Fine Tuning: A Comprehensive Analysis Across Applications. *arXiv*. http://arxiv.org/abs/2404.13506

Beddiar, D. R., Jahan, M. S., & Oussalah, M. (2021). Data expansion using back translation *and* paraphrasing for hate speech detection. *Online Social Networks and Media*, *24*, 100153. https://doi.org/10.1016/j.osnem.2021.100153

Brownlee, J. (2020). *Cost-Sensitive Learning for Imbalanced Classification*. Diakses pada tanggal 13 November 2025, dari https://machinelearningmastery.com/cost-sensitive-learning-for-imbalanced-classification/

Chen, H., Jiao, F., Li, X., Qin, C., Ravaut, M., Zhao, R., Xiong, C., & Joty, S. (2024). ChatGPT's One-year Anniversary: Are Open-Source Large Language Models Catching up? *arXiv*. http://arxiv.org/abs/2311.16989

Corbeil, J. P., & Ghavidel, H. A. (2021). Assessing the Eligibility of Backtranslated Samples Based on Semantic Similarity for the Paraphrase Identification Task. Dalam R. Mitkov & G. Angelova (Ed.), *International Conference Recent Advances in Natural Language Processing, RANLP* (hlm. 301–308). INCOMA Ltd. https://doi.org/10.26615/978-954-452-072-4_035

Coursera. (2025). *What Is Hugging Face?*. Coursera. Diakses pada tanggal 7 November 2025, dari https://www.coursera.org/articles/what-is-hugging-face?msockid=10092f6443516a5628f43a3f42d86b66

Desiani, A., Adrezo, M., Kresnawati, E. S., Ermatita, Akbar, M., & Hasibuan, M. S. (2023). Back Translation-EDA *and* Transformer for Hate Speech Classification in Indonesian. *2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS)*, 611–616. https://doi.org/10.1109/ICIMCIS60089.2023.10348979

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv*. http://arxiv.org/abs/2305.14314

Dong, G., Yuan, H., Lu, K., Li, C., Xue, M., Liu, D., Wang, W., Yuan, Z., Zhou, C., & Zhou, J. (2024). How Abilities in Large Language Models are Affected by Supervised Fine-tuning Data Composition. *arXiv*. http://arxiv.org/abs/2310.05492

Fadilah, N., & Priyanta, S. (2022). Automatic Essay Scoring Using Data Augmentation in Bahasa Indonesia. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *16*(4), 401–410. https://doi.org/10.22146/ijccs.76396

Fathuddin, M. A. H., Mandyartha, E. P., & Nurlaili, A. L. (2025). Penerapan Sentence-Bert dan Cosine Similarity untuk Pencarian Semantik Dokumen Skripsi dalam Format PDF. *Ranah Research : Journal of Multidisciplinary Research and Development*, *8*(1), 322–337. https://doi.org/10.38035/rrj.v8i1.1865

Finetune Llama 3.1 with Unsloth. (2024, Juli 23). Diakses pada tanggal 7 November 2025, dari https://unsloth.ai/blog/llama3-1

Hadi, M. U., Tashi, Q. Al, Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Al-Garadi, M. A., Hassan, S. Z., Shoman, M., Wu, J., Mirjalili, S., & Shah, M. (2023). *Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects. TechRxiv.* https://doi.org/10.36227/techrxiv.23589741.v1.

Hakiki, M., & Fatichah, C. (2025). Klasifikasi Kemampuan Mahasiswa Berdasarkan Automatic Essay Scoring terhadap Jawaban Essay Ujian Kompetensi dengan Metode Machine Learning. *Jurnal Indonesia : Manajemen Informatika dan Komunikasi (JIMIK)*, *6*(3), 1532–1546. https://doi.org/10.63447/jimik.v6i3.1325

Herwanto, G. B., Sari, Y., Prastowo, B. N., Riasetiawan, M., Bustoni, I. A., & Hidayatulloh, I. (2018). UKARA: A Fast *and* Simple Automatic Short Answer Scoring System for Bahasa Indonesia. *International Conference on Education and Psychology*, 48–53. https://doi.org/10.26499/iceap.v2i1.95

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv*. http://arxiv.org/abs/2106.09685

Huizenga, E., & Hu, M. (2024, Oktober 5). *Supervised Fine Tuning for Gemini LLM.* Google Cloud Blog. Diakses pada tanggal 13 November 2025, dari https://cloud.google.com/blog/products/ai-machine-learning/supervised-fine-tuning-for-gemini-llm

Humaira, R. (2025). *Automated Essay Scoring untuk Penilaian Jawaban Esai Bahasa Indonesia Dengan Indobert Embedding dan Feedforward Neural Network* [Skripsi, Universitas Sriwijaya]. https://repository.unsri.ac.id/168660/

Hussein, M. A., Hassan, H., & Nassef, M. (2019). Automated language essay scoring systems: A literature review. *PeerJ Computer Science*, *5*, e208. https://doi.org/10.7717/peerj-cs.208

Kaope, C., & Pristyanto, Y. (2023). The Effect of Class Imbalance Handling on Datasets Toward Classification Algorithm Performance. *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, *22*(2), 227–238. https://doi.org/10.30812/matrik.v22i2.2515

Khairani, U., Mutiawani, V., & Ahmadian, H. (2024). Pengaruh Tahapan Preprocessing Terhadap Model Indobert dan Indobertweet Untuk Mendeteksi

Emosi Pada Komentar Akun Berita Instagram. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *11*(4), 887–894. https://doi.org/10.25126/jtiik.1148315

Khoerunnisa, S. F., Surarso, B., & Kusumaningrum, R. (2025). Fine-tuning bidirectional encoder representations from transformers for the X social media personality detection. *IAES International Journal of Artificial Intelligence*, *14*(4), 3395–3403. https://doi.org/10.11591/ijai.v14.i4.pp3395-3403

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

Kinanti, N. L., & Qoiriah, A. (2021). Sistem Penilaian Otomatis Jawaban Esai Bahasa Indonesia Berdasarkan Kemiripan Kalimat Menggunakan Syntactic-Semantic Similarity. *Journal of Informatics and Computer Science*, *02*, 136–144. https://doi.org/10.26740/jinacs.v2n02.p136-144

Kotha, U. M., Gaddam, H., Siddenki, D. R., & Saleti, S. (2023). A comparison of various machine learning algorithms *and* execution of flask deployment on essay grading. *International Journal of Electrical and Computer Engineering*, *13*(3), 2990–2998. https://doi.org/10.11591/ijece.v13i3.pp2990-2998

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple *and* language independent subword tokenizer *and* detokenizer for Neural Text Processing. *arXiv*. http://arxiv.org/abs/1808.06226

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). 5 - Foundations of data imbalance *and* solutions for a data democracy. Dalam F. A. Batarseh & R. Yang (Ed.), *Data Democracy* (hlm. 83–106). Academic Press. https://doi.org/10.1016/B978-0-12-818366-3.00005-8

Lialin, V., Deshpande, V., Yao, X., & Rumshisky, A. (2024). Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning. *arXiv*. http://arxiv.org/abs/2303.15647

Lim, C. T., Bong, C. H., Wong, W. S., & Lee, N. K. (2021). A comprehensive Review of Automated Essay Scoring (AES) Research *and* Development. *Pertanika Journal of Science and Technology*, *29*(3), 1875–1899. https://doi.org/10.47836/pjst.29.3.27

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. (2022). Few-Shot Parameter-Efficient Fine-Tuning is Better *and* Cheaper than In-Context Learning. *arXiv*. http://arxiv.org/abs/2205.05638

Meta AI. (2024). *Llama 3 Model Cards and Prompt formats*. Diakses pada tanggal 7 November 2025, dari https://www.llama.com/docs/model-cards-and-prompt-formats/meta-llama-3/

Manchanda, J., Boettcher, L., Westphalen, M., & Jasser, J. (2025). The Open Source Advantage in Large Language Models (LLMs). *arXiv*. http://arxiv.org/abs/2412.12004

Mansha, I. (2025). Resource-Efficient Fine-Tuning of LLaMA-3.2-3B for Medical Chain-of-Thought Reasoning. *arXiv*. http://arxiv.org/abs/2510.05003

Meeradevi, Sowmya, B. J., & Swetha, B. N. (2024). Evaluating the machine learning models based on natural language processing tasks. *IAES International Journal of Artificial Intelligence*, *13*(2), 1952–1966. https://doi.org/10.11591/ijai.v13.i2.pp1954-1968

Meta AI. (2024, April 18). *Introducing Meta Llama 3: The most capable openly available LLM to date*. Diakses pada tanggal 15 November 2025, dari https://ai.meta.com/blog/meta-llama-3/

Mizumoto, A., & Eguchi, M. (2023). Exploring the potential of using an AI language model for automated essay scoring. *Research Methods in Applied Linguistics*, *2*(2), 100050. https://doi.org/10.1016/j.rmal.2023.100050

Mufiid, I., Lestanti, S., & Kholila, N. (2021). Aplikasi Penilaian Jawaban Esai Otomatis Menggunakan Metode Synonym Recognition dan Cosine Similarity Berbasis Web. *Jurnal Mnemonic*, *4*(2), 31–37. https://doi.org/10.36040/mnemonic.v4i2.4067

Norvig, P. (2007). *How to Write a Spelling Corrector*. Diakses pada tanggal 12 November 2025, dari https://norvig.com/spell-correct.html

Ormerod, C. M., Malhotra, A., & Jafari, A. (2021). Automated essay scoring using efficient transformer-based language models. *arXiv*. http://arxiv.org/abs/2102.13136

Pack, A., Barrett, A., & Escalante, J. (2024). Large language models *and* automated essay scoring of English language learner writing: Insights into validity *and* reliability. *Computers and Education: Artificial Intelligence*, *6*, 100234. https://doi.org/10.1016/j.caeai.2024.100234

Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024). The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges *and* Opportunities. *arXiv*. http://arxiv.org/abs/2408.13296

Pradani, K. A., & Suadaa, L. H. (2023). Automated Essay Scoring Menggunakan Semantic Textual Similarity Berbasis Transformer Untuk Penilaian Ujian Esai. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *10*(6), 1177–1184. https://doi.org/10.25126/jtiik.2023107338

Pratap, S., Aranha, A. R., Kumar, D., Malhotra, G., Iyer, A. P. N., & S.S., S. (2025). The fine art of fine-tuning: A structured review of advanced LLM fine-tuning techniques. *Natural Language Processing Journal*, *11*, 100144. https://doi.org/10.1016/j.nlp.2025.100144

Putra, I. F. (2020). Indonesian Essay Scoring using Bi-LSTM with Word Embedding Representation. Dalam *UKARA 1.0 Challenge*. https://github.com/ilhamfp/ukara-1.0-challenge

Radliński, Ł., Guściora, M., & Kocoń, J. (2025). Backtranslation *and* Paraphrasing in the LLM Era? Comparing Data Augmentation Methods for Emotion Classification. Dalam *Computational Science – ICCS 2025* (hlm. 3–17). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-97626-1_1

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., & Azam, S. (2024). A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues *and* Challenges. *IEEE Access*, *12*, 26839–26874. https://doi.org/10.1109/ACCESS.2024.3365742

Rajagede, R. A. (2021). Improving Automatic Essay Scoring for Indonesian Language using Simpler Model *and* Richer Feature. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 11–18. https://doi.org/10.22219/kinetik.v6i1.1196

Rajagede, R. A., & Hastuti, R. P. (2020). Automatic Short Answer Scoring for Bahasa Indonesia with Classifier Stacking. Dalam *UKARA 1.0 Challenge*. https://github.com/rianrajagede/ukara-challenge

Rajagede, R. A., & Hastuti, R. P. (2021). Stacking Neural Network Models for Automatic Short Answer Scoring. *IOP Conference Series: Materials Science and Engineering*, *1077*(1), 012013. https://doi.org/10.1088/1757-899x/1077/1/012013

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

Sanjaya, A., Setiawan, A. B., Mahdiyah, U., Farida, I. N., & Prasetyo, A. R. (2023). Pengukuran Kemiripan Makna Menggunakan Cosine Similarity dan Basis Data Sinonim Kata. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, *10*(4), 747–752. https://doi.org/10.25126/jtiik.2023106864

Septiandri, A. A., & Winatmoko, Y. A. (2020). UKARA 1.0 Challenge Track 1: Automatic Short-Answer Scoring in Bahasa Indonesia. *arXiv*. http://arxiv.org/abs/2002.12540

Septiandri, A. A., Winatmoko, Y. A., & Putra, I. F. (2020). Knowing Right from Wrong: Should We Use More Complex Models for Automatic Short-Answer Scoring in Bahasa Indonesia? Dalam N. S. Moosavi, A. Fan, V. Shwartz, G. Glavaš, S. Joty, A. Wang, & T. Wolf (Ed.), *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing* (hlm. 1–7). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.sustainlp-1.1

Siino, M., Tinnirello, I., & La Cascia, M. (2024). Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers *and* traditional classifiers. *Information Systems*, *121*, 102342. https://doi.org/10.1016/j.is.2023.102342

Song, Y., Zhu, Q., Wang, H., & Zheng, Q. (2024). Automated Essay Scoring *and* Revising Based on Open-Source Large Language Models. *IEEE Transactions on Learning Technologies*, *17*, 1880–1890. https://doi.org/10.1109/TLT.2024.3396873

Spelmen, V. S., & Porkodi, R. (2018). A Review on Handling Imbalanced Data. *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 1–11. https://doi.org/10.1109/ICCTCT.2018.8551020

Stryker, C. (2025, Mei 2). *What is Hugging Face?*. IBM. Diakses pada tanggal 6 November 2025, dari https://www.ibm.com/think/topics/hugging-face?mhsrc=ibmsearch_a&mhq=what%20is%20hugging%20face

Tamang, M. (2024, September 1). *Build Your Own Llama 3 Architecture from Scratch Using PyTorch* . Medium. Diakses pada tanggal 25 Februari 2026, dari https://pub.towardsai.net/build-your-own-llama-3-architecture-from-scratch-using-pytorch-2ce1ecaa901c

Tanaka, E. A., Christian, S., Anderies, & Chowanda, A. (2024). Evaluating Back Translation *and* Misspelling Correction Utilization on Indonesian AES. *2024 5th International Conference on Artificial Intelligence and Data Sciences, AiDAS 2024 - Proceedings*, 1–5. https://doi.org/10.1109/AiDAS63860.2024.10730568

Tunstall, L., Werra, L. von, & Wolf, T. (2022). *Natural Language Processing with Transformers*. O'Reilly Media. Diakses pada tanggal 25 Februari 2026, dari https://books.google.co.id/books?id=nzxbEAAAQBAJ

Universitas Gadjah Mada. (2019). *UKARA 1.0 Challenge Dataset*. Simpan.ugm.ac.id. https://simpan.ugm.ac.id/s/sRHfHYpSqeu9tDs

Unsloth. (2025). *Unsloth Documentation*. Diakses pada tanggal 7 November 2025, dari https://docs.unsloth.ai/

Unsloth. (2024). *Unsloth/Meta-Llama-3.1-8B-Instruct · Hugging Face*. Diakses pada tanggal 7 November 2025, dari https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct

Vaniukov, S. (2023, November 29). *How to Build a Large Language Model: Step-by-Step Guide*. Diakses pada tanggal 7 November 2025, dari https://www.softermii.com/blog/how-to-build-a-large-language-model-step-by-step-guide

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *arXiv*. http://arxiv.org/abs/1706.03762

Veena, G., Rajendran, A., & Gupta, D. (2025). Extracting Triplets from Domain-specific Texts using Fine-Tuned LLaMA Models: A Comparative Study. *Procedia Computer Science*, *258*, 3750–3759. https://doi.org/10.1016/j.procs.2025.04.630

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., & Hajishirzi, H. (2023). Self-Instruct: Aligning Language Models with Self-Generated Instructions. *arXiv*. http://arxiv.org/abs/2212.10560