

**PERBANDINGAN PELABELAN SENTIMEN
BERBANTUAN *GENERATIVE AI***



Disusun Oleh:

Nama : Arya Eggy Aditya
NIM : 22523204

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PERBANDINGAN PELABELAN SENTIMEN

BERBANTUAN GENERATIVE AI

TUGAS AKHIR



N a m a : Arya Eggy Aditya
NIM : 22523204

Yogyakarta, 28 April 2026

Pembimbing,



(Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PERBANDINGAN PELABELAN SENTIMEN BERBANTUAN *GENERATIVE AI*

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 13 Juni 2026

Tim Penguji

Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs.

Anggota 1

Dr. Ahmad Luthfi, S.Kom., M.Kom.

Anggota 2

Ir. Chandra Kusuma Dewa, S.Kom.,

M.Cs., Ph.D.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Ir. Dhomas Hatta Fucholi, S.T., M.Eng., Ph.D., IPM., ASEAN Eng.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Arya Eggy Aditya

NIM : 22523204

Tugas akhir dengan judul:

**PERBANDINGAN PELABELAN SENTIMEN
BERBANTUAN *GENERATIVE AI***

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 19 Mei 2026



(Arya Eggy Aditya)

HALAMAN PERSEMBAHAN

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat dan karunia-Nya sehingga tugas akhir ini dapat terselesaikan. Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW sebagai suri tauladan bagi seluruh umat manusia. Laporan tugas akhir ini penulis persembahkan kepada Bapak dan Ibu tercinta, atas segala doa, kasih sayang, dan pengorbanan tulus yang tidak pernah berhenti diberikan kepada penulis. Penulis juga ingin mengucapkan terima kasih kepada Ibu Dr. Nur Wijyaning Rahayu, S.Kom., M.Cs., selaku dosen pembimbing yang telah membimbing penulis dengan penuh kesabaran dan ketulusan. Serta kepada seluruh pihak yang telah turut membantu dan mendukung penyelesaian tugas akhir ini, penulis ucapkan terima kasih yang sebesar-besarnya. Semoga tugas akhir ini dapat menjadi amal yang bermanfaat dan menjadi referensi yang berguna bagi penelitian selanjutnya..

HALAMAN MOTO

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

(QS. Al-Baqarah: 286)

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala limpahan rahmat dan karunia-Nya, serta shalawat dan salam yang senantiasa tercurahkan kepada junjungan Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul "Perbandingan Pelabelan Sentimen Berbantuan *Generative AI*". Penyelesaian tugas akhir ini tentunya tidak lepas dari bimbingan, dukungan, dan doa dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak dan Ibu tercinta atas segala doa, kasih sayang, dukungan, dan pengorbanan yang tidak pernah berhenti diberikan kepada penulis.
2. Keluarga besar penulis, yang selalu mendoakan dan memberikan semangat.
3. Ibu Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs., selaku dosen pembimbing yang telah meluangkan waktu, pikiran, dan tenaga untuk membimbing penulis dengan penuh kesabaran selama penyusunan tugas akhir.
4. Bapak Dr. Feri Wijayanto, S.T., M.T., selaku dosen Informatika Universitas Islam Indonesia yang telah memberikan arahan, saran, dan dukungan selama proses penelitian berlangsung.
5. Seluruh dosen Program Studi Informatika Universitas Islam Indonesia atas ilmu dan pengalaman yang telah diberikan selama masa perkuliahan.
6. Teman-teman dekat penulis atas semangat, dukungan, dan kebersamaan yang sangat berarti selama proses penelitian ini.
7. Seluruh teman-teman Informatika Universitas Islam Indonesia angkatan 2022 (Invacy) atas kebersamaan, solidaritas, dan kenangan yang telah terukir selama masa perkuliahan.
8. Seluruh pihak yang telah membantu dalam penyelesaian penelitian ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan tugas akhir ini dapat bermanfaat untuk ke depannya.

Yogyakarta, 19 Mei 2026



(Arya Eggy Aditya)

SARI

Analisis sentimen terhadap respons mahasiswa merupakan pendekatan yang dapat digunakan untuk mengevaluasi efektivitas platform pembelajaran digital. Namun, proses analisis secara manual menjadi tidak praktis seiring bertambahnya volume data, sehingga diperlukan pendekatan komputasional yang lebih efisien. Penelitian ini bertujuan untuk membandingkan performa model *Generative AI* dalam melakukan pelabelan sentimen terhadap respons mahasiswa mengenai penggunaan Google Classroom. Penelitian ini menguji tiga model Large Language Model (LLM) secara *zero-shot*, yaitu GPT-4.1, GPT-4o, dan Claude Haiku 4.5, serta satu model berbasis *fine-tuning*, yaitu IndoBERT. Eksperimen dilakukan pada dua kondisi dataset, yaitu *raw* dan *preprocessed*, dengan masing-masing model dijalankan sebanyak tiga kali dan performanya dihitung sebagai nilai rata-rata dari ketiga run beserta standar deviasi sebagai indikator konsistensi. Evaluasi performa dilakukan menggunakan metrik *accuracy* dan *macro average F1-score*. Hasil penelitian menunjukkan Claude Haiku 4.5 mencatatkan performa terbaik dengan akurasi 85,59% dan *macro F1-score* 0,79 pada data *preprocessed*. GPT-4o dan GPT-4.1 juga menunjukkan hasil yang baik, masing-masing dengan akurasi terbaik 80,00% dan 78,92%. IndoBERT mencatatkan performa terendah dengan akurasi 76,22% dan *macro F1-score* 0,68 yang stagnan pada kedua kondisi dataset. Secara keseluruhan, ketiga model LLM yang diuji secara *zero-shot* melampaui performa IndoBERT yang telah melalui proses *fine-tuning*, mengindikasikan bahwa kapasitas pemahaman bahasa LLM skala besar cukup kuat untuk tugas analisis sentimen bahasa Indonesia tanpa pelatihan domain khusus.

Kata kunci: analisis sentimen, *generative AI*, *large language model*, *zero-shot*, IndoBERT

GLOSARIUM

BERT	Bidirectional Encoder Representations from Transformers, sebuah model <i>pre-trained</i> berbasis arsitektur Transformer yang mampu memahami konteks kata secara dua arah
<i>Generative AI</i>	Model yang mampu menghasilkan konten baru seperti gambar, teks, atau kode berdasarkan pola yang telah dipelajari
Ground Truth	Label data yang ditetapkan secara manual sebagai acuan kebenaran dalam evaluasi model
Inferensi	Proses penggunaan model untuk menghasilkan prediksi pada data baru

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI	viii
GLOSARIUM	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian Secara Umum	4
1.7 Sistematika Penulisan Laporan	5
BAB II LANDASAN TEORI	6
2.1 Kontekstualisasi Masalah.....	6
2.2 Analisis Sentimen	7
2.3 <i>Generative AI</i>	8
2.4 Peran Large Language Model dalam Analisis Sentimen	8
BAB III METODOLOGI PENELITIAN.....	10
3.1 Persiapan Dataset	10
3.2 Pelabelan Data Manual (<i>Ground Truth</i>)	11
3.3 <i>Data Preprocessing</i>	11
3.4 Inferensi dan Evaluasi Model Pendekatan <i>Zero-shot</i>	13
3.4.1 Zero-shot	13
3.4.2 Inferensi	14

3.4.3	Evaluasi Model	15
3.5	Inferensi dan Evaluasi Model Pendekatan <i>Fine-tuning</i>	18
3.5.1	Fine-tuning Model.....	18
3.5.2	Inferensi	21
3.5.3	Evaluasi Model	21
3.6	Komparasi Kinerja Model.....	22
BAB IV HASIL DAN PEMBAHASAN.....		23
4.1	Persiapan Dataset	23
4.2	Pelabelan Data Manual	25
4.3	<i>Data Preprocessing</i>	25
4.4	Inferensi dan Evaluasi Model Pendekatan <i>Zero-shot</i>	27
4.4.1	System Prompt	27
4.4.2	Inferensi dan Hasil Prediksi Model GPT-4o.....	28
4.4.3	Inferensi dan Hasil Prediksi Model GPT-4.1	30
4.4.4	Inferensi dan Hasil Prediksi Model Claude Haiku 4.5.....	30
4.4.5	Evaluasi Model GPT-4o.....	32
4.4.6	Evaluasi Model GPT-4.1.....	35
4.4.7	Evaluasi Model Claude Haiku 4.5	38
4.5	Inferensi dan Evaluasi Model Pendekatan Fine-tuning	41
4.5.1	Persiapan Model dan Dataset.....	41
4.5.2	Proses Fine-tuning.....	42
4.5.3	Inferensi dan Hasil Prediksi Model IndoBERT	44
4.5.4	Evaluasi Model IndoBERT	46
4.6	Komparasi Model.....	47
4.7	Generalisasi Desain Prompt	49
BAB V KESIMPULAN DAN SARAN.....		51
5.1	Kesimpulan	51
5.2	Saran.....	51
DAFTAR PUSTAKA.....		52
LAMPIRAN		55

DAFTAR TABEL

Tabel 3.1 Spesifikasi teknis model LLM.....	14
Tabel 3.2 Pembagian dataset SmSA	19
Tabel 3.3 Konfigurasi DataLoader	20
Tabel 3.4 <i>Hyperparameter</i> pelatihan IndoBERT.....	21
Tabel 4.1 Contoh perubahan dari setiap langkah <i>preprocessing</i>	27
Tabel 4.2 Hasil inferensi GPT-4o pada kedua dataset.....	29
Tabel 4.3 Hasil inferensi GPT-4.1 pada kedua dataset.....	30
Tabel 4.4 Hasil inferensi Claude Haiku 4.5 pada kedua dataset.....	31
Tabel 4.5 Hasil evaluasi <i>mean</i> dan standar deviasi model GPT-4o.....	32
Tabel 4.6 Hasil evaluasi <i>mean</i> dan standar deviasi model GPT-4.1	35
Tabel 4.7 Hasil evaluasi <i>mean</i> dan standar deviasi model Claude Haiku 4.5	38
Tabel 4.8 Hasil <i>fine-tuning</i> model	44
Tabel 4.9 Hasil evaluasi <i>mean</i> dan standar deviasi model IndoBERT	46
Tabel 4.10 Performa lengkap setiap model diurutkan dari performa tertinggi.....	49
Tabel 4.11 Keterangan <i>placeholder</i> templat <i>prompt</i>	50

DAFTAR GAMBAR

Gambar 3.1 Tahapan penelitian	10
Gambar 3.2 <i>Confusion matrix</i>	15
Gambar 4.1 Dataset respons mahasiswa.....	23
Gambar 4.2 Halaman repositori GitHub IndoNLU	24
Gambar 4.3 Sampel dataset <i>train</i> pada SmSA.....	24
Gambar 4.4 Dataset hasil pelabelan manual	25
Gambar 4.5 Kode program tahap <i>case folding</i>	26
Gambar 4.6 Kode program tahap <i>repeated character normalization</i>	26
Gambar 4.7 Kode program tahap <i>emoji removal</i>	26
Gambar 4.8 Kode program tahap <i>whitespace cleaning</i>	26
Gambar 4.9 <i>System prompt</i> penelitian	28
Gambar 4.10 Kode fungsi klasifikasi model GPT	28
Gambar 4.11 Kode proses klasifikasi model GPT dan penyimpanan hasil.....	29
Gambar 4.12 Kode fungsi klasifikasi model Claude Haiku 4.5	30
Gambar 4.13 Kode proses klasifikasi model Claude Haiku 4.5 dan penyimpanan hasil	31
Gambar 4.14 Kode proses evaluasi model.....	32
Gambar 4.15 Evaluasi <i>best run</i> (#1) GPT-4o pada dataset <i>raw</i>	33
Gambar 4.16 Evaluasi <i>best run</i> (#3) GPT-4o pada dataset <i>preprocessed</i>	33
Gambar 4.17 <i>Confusion matrix run</i> terbaik GPT-4o dataset <i>raw</i>	34
Gambar 4.18 <i>Confusion matrix run</i> terbaik GPT-4o dataset <i>preprocessed</i>	35
Gambar 4.19 Evaluasi <i>best run</i> (#3) GPT-4.1 pada dataset <i>raw</i>	36
Gambar 4.20 Evaluasi <i>best run</i> (#1) GPT-4.1 pada dataset <i>preprocessed</i>	36
Gambar 4.21 <i>Confusion matrix run</i> terbaik GPT-4.1 dataset <i>raw</i>	37
Gambar 4.22 <i>Confusion matrix run</i> terbaik GPT-4.1 dataset <i>preprocessed</i>	37
Gambar 4.23 Evaluasi <i>best run</i> (#1) Claude Haiku 4.5 pada dataset <i>raw</i>	39
Gambar 4.24 Evaluasi <i>best run</i> (#2) Claude Haiku 4.5 pada dataset <i>preprocessed</i>	39
Gambar 4.25 <i>Confusion matrix run</i> terbaik Claude Haiku 4.5 dataset <i>raw</i>	40
Gambar 4.26 <i>Confusion matrix run</i> terbaik Claude Haiku 4.5 dataset <i>preprocessed</i>	40
Gambar 4.27 Kode impor <i>library</i> yang digunakan.....	41
Gambar 4.28 Penetapan <i>random seed</i>	41
Gambar 4.29 Kode pemuatan model dan <i>tokenizer</i> IndoBERT	42
Gambar 4.30 Kode persiapan dataset dan DataLoader	42

Gambar 4.31 Kode proses tahap <i>fine-tuning</i>	43
Gambar 4.32 Kode penyimpanan model hasil <i>fine-tuning</i>	44
Gambar 4.33 Kode <i>load</i> model hasil <i>fine-tuning</i> untuk inferensi.....	44
Gambar 4.34 Kode proses inferensi IndoBERT	45
Gambar 4.35 Kode penyimpanan hasil prediksi IndoBERT	45
Gambar 4.36 Evaluasi <i>run</i> IndoBERT	46
Gambar 4.37 <i>Confusion matrix</i> IndoBERT	47
Gambar 4.38 Templat <i>prompt</i> analisis sentimen	50

BAB I PENDAHULUAN

1.1 Latar Belakang

Umpan balik atau *feedback* merupakan suatu informasi yang diberikan sebagai tanggapan terhadap suatu tindakan, kinerja, ataupun hasil dari sebuah proses. Tujuan utama dari pemberian umpan balik adalah untuk mengevaluasi dan menyediakan data yang dapat digunakan untuk perbaikan, sehingga dapat memodifikasi pemikiran serta perilaku ke arah yang lebih baik (Prihandini, 2022). Pada hakikatnya, umpan balik adalah sebuah proses komunikasi yang menciptakan siklus interaktif antara pemberi dan penerima, yang berperan penting dalam berbagai bidang untuk meningkatkan kualitas dan performa (Prasetio, 2022).

Penerapan umpan balik dapat ditemukan dalam berbagai konteks. Di dunia bisnis, terdapat umpan balik pelanggan (*customer feedback*) yang berfokus pada kepuasan terhadap produk atau layanan (Hendrayani, Basir, Abidin, & Utomo, 2024). Dalam lingkungan kerja, ada umpan balik kinerja (*performance feedback*) untuk mengevaluasi produktivitas dan hasil kerja (Wahyudi, Gama, Kadek, & Pridayanti, 2024). Selain itu, ada juga umpan balik akademik (*academic feedback*) yang terjadi dalam proses belajar-mengajar antara pengajar dan siswa (Molina-Moreira, Velásquez-Orellana, Zambrano-Murillo, & Zambrano-Villamil, 2023).

Dalam konteks akademik, umpan balik dari mahasiswa memegang peranan yang sangat penting dan fundamental. Mahasiswa merupakan komponen krusial di tingkat perguruan tinggi yang masukannya bersifat konstruktif dan berkontribusi langsung pada pengembangan institusi maupun para pengajar (Baihaqi & Munandar, 2023). Sejalan dengan hal tersebut, umpan balik dari mahasiswa juga dipandang sebagai sumber informasi yang sangat diperlukan dan dapat dimanfaatkan oleh para pengajar untuk meningkatkan kualitas prosedur pembelajaran dan kegiatan akademik secara keseluruhan (Dalipi, Zdravkova, & Ahlgren, 2021). Dengan demikian, menampung dan memahami aspirasi mahasiswa melalui umpan balik menjadi langkah esensial bagi institusi akademik untuk terus berevolusi dan meningkatkan mutu layanannya.

Umpan balik mahasiswa yang terkumpul dalam jumlah besar kerap menjadi tantangan tersendiri bagi institusi akademik. Ketika volume data terus bertambah, pendekatan manual untuk menelaah setiap respons satu per satu menjadi tidak efisien dari segi waktu maupun tenaga. Kondisi ini mendorong kebutuhan akan solusi otomatis yang mampu mengolah data

tersebut secara efisien. Beberapa pendekatan telah tersedia untuk keperluan ini, mulai dari Word Cloud yang memberikan gambaran visual frekuensi kata, hingga *topic modeling* yang mengelompokkan komentar berdasarkan tema. Namun, apabila tujuannya adalah mengklasifikasikan opini menjadi kelompok sentimen yang eksplisit, maka analisis sentimen merupakan pendekatan yang paling tepat.

Analisis sentimen bekerja dengan cara mendeteksi dan mengekstraksi opini dari suatu teks, lalu mengklasifikasikannya menjadi kategori sentimen positif, negatif, atau netral (Hidayat & Pramudita, 2024). Teknik yang digunakan dalam proses ini terus berkembang dari waktu ke waktu. Pada tahap awal, pendekatan yang dominan adalah metode berbasis aturan seperti *lexicon-based*, yang bekerja dengan merujuk pada kamus kata bersentimen. Pendekatan berikutnya memanfaatkan algoritma Naive Bayes dan Support Vector Machine yang merupakan metode *machine learning* untuk mengenali pola sentimen dari data berlabel. Perkembangan berlanjut dengan kemunculan *deep learning* berbasis *neural network*, yang kemudian berevolusi menjadi arsitektur Transformer. Puncak perkembangan ini ditandai dengan hadirnya Large Language Models (LLM) seperti Bidirectional Encoder Representations from Transformers (BERT) dan Generative Pre-trained Transformer (GPT), yang mampu menangkap konteks linguistik secara lebih mendalam untuk menghasilkan klasifikasi yang lebih akurat (Gupta, Ranjan, & Singh, 2023).

Dalam ekosistem Bahasa Indonesia, IndoBERT menjadi salah satu model yang paling banyak diadopsi. Model ini diperkenalkan pada tahun 2020 dan dibangun dengan korpus besar bahasa Indonesia yang dikenal sebagai Indo4B, sehingga memiliki pemahaman yang baik terhadap konteks bahasa lokal (Wilie et al., 2020). Agar dapat diaplikasikan pada tugas analisis sentimen, IndoBERT memerlukan proses *fine-tuning* menggunakan dataset berlabel yang relevan. Meski pendekatan ini terbukti memberikan hasil yang baik, terdapat konsekuensi yang perlu dipertimbangkan. Ukuran arsitekturnya yang besar membuat proses pelatihan membutuhkan waktu yang cukup panjang dan sumber daya komputasi yang tidak sedikit (Sayeed, Mohan, & Muthu, 2023).

Berbeda dengan pendekatan tersebut, LLM berbasis *Generative AI* seperti GPT menawarkan jalur alternatif melalui kemampuan *zero-shot learning*, yaitu kemampuan melakukan klasifikasi sentimen tanpa memerlukan pelatihan eksplisit pada data tugas yang dituju (Wang et al., 2024). Kemampuan ini terbukti menghasilkan performa yang kompetitif, bahkan dalam beberapa kasus mampu melampaui model yang telah melalui *fine-tuning* seperti BERT (Wang et al., 2024), (Lossio-Ventura et al., 2024). Hal ini menjadikan *Generative AI*

sebagai alternatif yang menarik karena menghilangkan ketergantungan pada data berlabel dalam jumlah besar sekaligus meniadakan proses pelatihan yang memakan waktu.

Meski demikian, kajian yang secara langsung membandingkan model-model *Generative AI* seperti GPT dan Claude dengan model lokal yang di-*fine-tuning* seperti IndoBERT dalam konteks bahasa Indonesia masih belum banyak dilakukan. Kesenjangan ini mencakup pemahaman tentang sejauh mana efektivitas masing-masing pendekatan dalam menghasilkan klasifikasi yang akurat dan seimbang di seluruh kelas sentimen, khususnya diukur melalui akurasi dan *macro average F1-score*. Maka dari itu, penelitian ini dilakukan guna menjawab kesenjangan tersebut dengan membandingkan secara langsung performa IndoBERT yang telah di-*fine-tuning* dengan beberapa model *Generative AI* populer dalam tugas pelabelan sentimen pada konteks akademik.

1.2 Rumusan Masalah

Analisis sentimen pada umpan balik mahasiswa masih bergantung pada pendekatan yang membutuhkan sumber daya besar, baik melalui analisis manual yang tidak efisien seiring meningkatnya volume data, maupun proses *fine-tuning* model seperti IndoBERT yang memerlukan data berlabel, waktu pelatihan, dan komputasi yang signifikan. Di sisi lain, studi perbandingan antara model lokal berbasis *fine-tuning* dengan model *Generative AI* berbasis *zero-shot* dalam konteks bahasa Indonesia masih terbatas. Dari permasalahan tersebut, penelitian ini merumuskan pertanyaan penelitian berupa “*Bagaimana perbandingan performa model Generative AI secara zero-shot dengan model IndoBERT yang di-fine-tuning dalam tugas pelabelan sentimen pada data umpan balik mahasiswa?*”

1.3 Batasan Masalah

Agar penelitian dapat berjalan secara terarah, terdapat beberapa batasan dan lingkup yang ditetapkan sebagai berikut:

- a. Klasifikasi sentimen dibatasi pada tiga kelas, yaitu positif, netral, dan negatif
- b. Evaluasi utama performa dilakukan menggunakan metrik *accuracy* dan *macro average F1-score*

1.4 Tujuan Penelitian

Penelitian bertujuan untuk membandingkan performa model *Generative AI* (GPT-4.1, GPT-4o, Claude Haiku 4.5) secara *zero-shot* dengan model IndoBERT yang di-*fine-tuning* dalam tugas pelabelan sentimen pada data umpan balik mahasiswa. Sebagai tujuan pendukung, penelitian ini juga menganalisis pengaruh *preprocessing* data uji terhadap performa masing-masing model sebagai temuan tambahan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan mampu menjadi acuan dalam pemilihan model yang paling efektif dan efisien untuk pelabelan sentimen pada konteks akademik. Di samping itu, penelitian ini diharapkan mampu menambah referensi mengenai perbandingan pendekatan *zero-shot* berbasis *Generative AI* dengan model lokal berbasis *fine-tuning* dalam konteks bahasa Indonesia, yang saat ini masih terbatas.

1.6 Metodologi Penelitian Secara Umum

Penelitian ini dilaksanakan melalui beberapa tahapan sebagai berikut:

a. Studi Literatur

Tahap awal untuk mencari dan menelaah jurnal serta bacaan ilmiah yang relevan mencakup konsep analisis sentimen, perkembangan metode klasik, serta modern seperti LLM dan *Generative AI*.

b. Persiapan Dataset

Proses penyiapan dan pengolahan seluruh dataset yang akan digunakan dalam eksperimen.

c. Eksperimen

Proses inferensi setiap model pada masing-masing kondisi dataset untuk menghasilkan prediksi label sentimen.

d. Evaluasi

Pengukuran performa dari hasil setiap model menggunakan metrik yang ditetapkan (*accuracy*, *precision*, *recall*, dan *f1-score*).

e. Analisis dan Kesimpulan

Proses analisis hasil evaluasi untuk membandingkan performa antar model dan mengambil kesimpulan akhir penelitian.

1.7 Sistematika Penulisan Laporan

Laporan tugas akhir ini disusun menjadi lima bab utama. Pembagian ini bertujuan untuk menyajikan alur informasi yang terstruktur sehingga isi penelitian mudah dipahami. Berikut adalah rincian sistematikanya:

a. BAB I PENDAHULUAN

Pada bagian pendahuluan ini membahas latar belakang, pendefinisian ruang lingkup masalah, tujuan, manfaat, tahap penelitian secara umum, serta sistematika penulisan.

b. BAB II LANDASAN TEORI

Bagian ini membahas teori yang relevan dengan penelitian, mencakup konsep analisis sentimen beserta perkembangan metodenya serta model-model yang digunakan dalam eksperimen.

c. BAB III METODOLOGI PENELITIAN

Bab ini membahas seluruh langkah teknis eksperimen secara sistematis. Pembahasannya mencakup persiapan dan pemrosesan dataset, skenario inferensi model (GPT-4.1, GPT-4o, dan Claude Haiku 4.5), hingga penetapan metrik evaluasi kinerja.

d. BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan temuan eksperimen yang telah dilakukan dan menganalisis perbandingan performa antar model berdasarkan metrik yang telah ditetapkan.

e. BAB V KESIMPULAN DAN SARAN

Bagian penutup ini menyajikan ringkasan temuan serta saran guna menjadi pertimbangan bagi penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Kontekstualisasi Masalah

Feedback mahasiswa merupakan salah satu sumber data yang krusial bagi institusi pendidikan, karena dapat dimanfaatkan untuk mengevaluasi prosedur dan kualitas proses belajar mengajar (Dalipi et al., 2021). Bentuknya beragam, mulai dari penilaian kinerja dosen hingga ulasan penggunaan platform pembelajaran digital seperti Google Classroom. Dengan menganalisis *feedback* ini secara sistematis, area yang memerlukan perbaikan dalam proses pembelajaran dapat diidentifikasi secara lebih tepat (Baihaqi & Munandar, 2023). Namun, volume data yang terus bertambah membuat analisis *feedback* secara manual menjadi tidak efektif dan efisien (Kastrati, Dalipi, Imran, Nuci, & Wani, 2021). Analisis sentimen hadir sebagai solusi komputasional yang mampu mengklasifikasikan orientasi opini dalam suatu teks ke dalam kategori tertentu seperti positif, netral, atau negatif secara otomatis (Hidayat & Pramudita, 2024). Dalam praktiknya, pendekatan yang digunakan untuk analisis sentimen cukup beragam.

Pendekatan berbasis leksikon, seperti yang diimplementasikan oleh Arifin & Febryanto (2025) menggunakan metode VADER Lexicon pada data ulasan Aplikasi Merdeka Mengajar, terbukti efisien dalam hal kecepatan komputasi. Akan tetapi, pendekatan ini memiliki keterbatasan dalam menangani variasi bahasa lokal, ironi, dan sarkasme. Pada pendekatan *machine learning*, Baihaqi & Munandar (2023) menerapkan algoritma Naïve Bayes untuk mengklasifikasikan sentimen pada komentar mahasiswa terkait kinerja perguruan tinggi. Setelah proses ekstraksi fitur, tingkat akurasi yang dicapai sebesar 74%. Meskipun relatif cepat dan mudah diimplementasikan, metode ini rentan terhadap kesalahan jika pemilihan fitur kurang tepat dan kurang optimal untuk teks yang panjang, sebagaimana dilaporkan oleh Razaq, Nurjanah, & Nurrahmi (2023).

Dalam penelitian yang sama, Baihaqi & Munandar (2023) turut mengevaluasi IndoBERT sebagai representasi pendekatan *deep learning* berbasis arsitektur Transformer. Hasilnya menunjukkan bahwa IndoBERT mampu melampaui performa Naïve Bayes dengan akurasi sebesar 85%, yang mengindikasikan keunggulan mekanisme *attention* dalam memahami konteks semantik teks. Meski demikian, model berbasis BERT tetap memiliki kelemahan, yaitu kebutuhan sumber daya komputasi yang tinggi (Sayeed et al., 2023) serta risiko *overfitting*

yang signifikan apabila dilatih pada dataset berukuran kecil (Wu, Jin, Shi, Liang, & Zhan, 2024).

Dari uraian di atas, tampak bahwa setiap pendekatan membawa konsekuensi tersendiri dalam hal akurasi, efisiensi, dan kebutuhan sumber daya. Perkembangan terkini menunjukkan bahwa LLM berbasis *zero-shot* mulai menunjukkan performa yang kompetitif terhadap model *deep learning* yang telah dilatih secara khusus. Hidayaturrehman & Prawira (2024) dalam penelitian perbandingan LLM dan model tradisional untuk analisis sentimen bahasa Indonesia menemukan bahwa GPT-3.5 dengan pendekatan *zero-shot* mencapai *F1-score* sebesar 0,83, melampaui IndoBERT yang hanya mencapai 0,76 pada dataset yang sama. Temuan ini mengindikasikan bahwa LLM berpotensi menjadi alternatif yang layak dibandingkan model *fine-tuned* tanpa memerlukan data latih sama sekali, khususnya dalam konteks teks berbahasa Indonesia. Oleh karena itu, penelitian ini bertujuan untuk meneliti lebih lanjut sejauh mana perbandingan performa antara kedua pendekatan tersebut, khususnya pada domain *feedback* mahasiswa.

2.2 Analisis Sentimen

Analisis sentimen adalah teknik pengumpulan dan pengkajian opini maupun persepsi individu terhadap suatu entitas, baik berupa topik, produk, maupun layanan. Pendekatan yang kerap dikenal dengan istilah *opinion mining* ini merupakan cabang dari Natural Language Processing (NLP) yang difokuskan untuk mengenali dan mengekstrak muatan subjektif dalam teks secara komputasional. Pada praktiknya, *output* dari proses ini umumnya dikategorikan menjadi sentimen positif, netral, dan negatif (Wankhade, Rao, & Kulkarni, 2022).

Dalam implementasinya, analisis sentimen dapat dilakukan pada beberapa tingkatan granularitas: tingkat dokumen memberikan satu polaritas untuk keseluruhan teks, tingkat kalimat menganalisis setiap kalimat secara individual, serta tingkat frasa dan aspek memungkinkan analisis yang lebih terperinci berdasarkan aspek-aspek spesifik yang disebutkan dalam teks.

Analisis sentimen dapat dilakukan menggunakan berbagai pendekatan yang secara umum terbagi menjadi tiga kategori: *machine learning* konvensional, *deep learning*, dan *ensemble learning*. Pendekatan *machine learning* seperti Naïve Bayes dan SVM mengandalkan fitur statistik dari teks, sementara pendekatan *deep learning* seperti LSTM dan BERT memanfaatkan representasi kontekstual. Adapun *ensemble learning* menggabungkan prediksi dari beberapa model untuk meningkatkan performa keseluruhan (Tan, Lee, & Lim, 2023).

Meski terus berkembang, analisis sentimen masih menghadapi sejumlah tantangan seperti teks yang tidak terstruktur, mengandung sarkasme, atau ditulis dalam gaya informal seperti unggahan media sosial kerap menyebabkan model gagal menangkap sentimen yang sesungguhnya. Di samping itu, model yang dilatih pada satu bahasa atau budaya tertentu cenderung tidak efektif ketika diterapkan pada konteks bahasa lain karena perbedaan idiom dan ekspresi yang signifikan. Tantangan ini menjadi sangat relevan dalam konteks bahasa Indonesia, yang kaya akan variasi ragam bahasa informal, bahasa daerah, dan ekspresi slang dalam komunikasi sehari-hari.

2.3 *Generative AI*

Generative AI (GenAI) merupakan salah satu cabang kecerdasan buatan yang difokuskan untuk memproduksi konten sintesis baru, baik berupa teks, gambar, maupun audio, melalui teknik *generative modeling*. Pendekatan ini berbeda dengan model diskriminatif yang sekadar memprediksi label dari sebuah input. Model generatif bekerja dengan cara mempelajari distribusi data latih untuk menciptakan *output* baru yang karakteristiknya menyerupai data asli.

Sebagai salah satu *subset* dari *Generative AI*, Large Language Model (LLM) adalah model berbasis Transformer yang dilatih menggunakan korpus teks berukuran masif agar mampu memahami dan menghasilkan teks layaknya manusia. Keunggulan utama dari LLM adalah kemampuannya dalam mengeksekusi instruksi spesifik tanpa membutuhkan data latih tambahan, atau biasa disebut dengan inferensi *zero-shot* (García-Peñalvo & Vázquez-Ingelmo, 2023). Karakteristik inilah yang membuat pemanfaatan LLM menjadi sangat relevan untuk melakukan tugas analisis sentimen secara langsung tanpa harus melewati tahapan *fine-tuning*.

2.4 Peran Large Language Model dalam Analisis Sentimen

Era Large Language Model (LLM) dimulai sejak kehadiran GPT-3 pada tahun 2020, yang membuktikan bahwa peningkatan skala parameter secara masif dapat menghasilkan kemampuan pemahaman dan generasi bahasa alami yang jauh melampaui model-model sebelumnya (Minaee et al., 2025). Perkembangan berikutnya ditandai oleh kemunculan ChatGPT pada akhir 2022, sebuah model yang dibangun di atas GPT-3.5 dan dilatih menggunakan teknik Reinforcement Learning from Human Feedback (RLHF) sehingga mampu mengikuti instruksi dalam bahasa alami secara lebih konsisten (Gao et al., 2023). Perilisan GPT-4 pada 2023 melanjutkan tren ini dengan menunjukkan peningkatan performa pada berbagai tugas kompleks sekaligus memperluas kapabilitas model ke modalitas di luar

teks (Gao et al., 2023). Hingga kini, perkembangan tersebut terus berlanjut dengan hadirnya beragam model generasi terkini dari berbagai organisasi, di antaranya seperti seri GPT dari OpenAI, Claude dari Anthropic, dan Gemini dari Google.

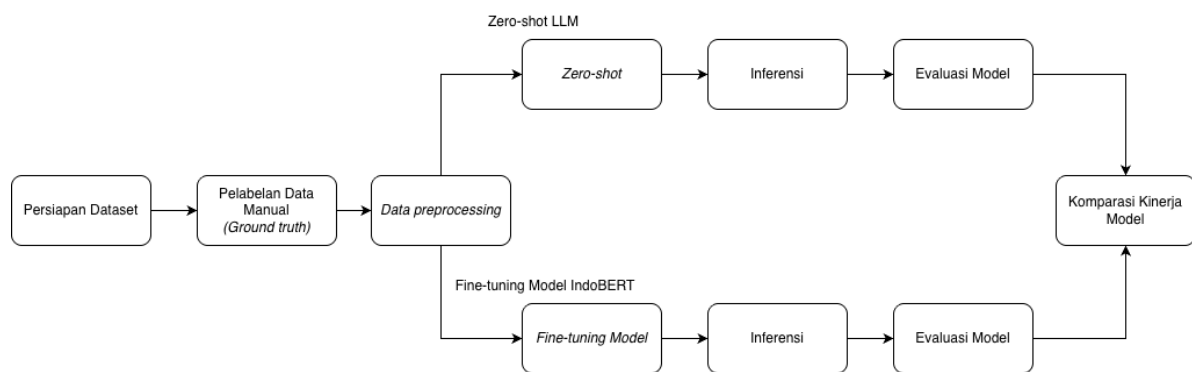
Perkembangan model-model LLM tersebut mendorong penerapan *Generative AI* untuk tugas analisis sentimen ke berbagai domain, mulai dari ulasan konsumen hingga survei dan *feedback* mahasiswa di lingkungan pendidikan (Parker, Anderson, Stone, & Oh, 2025). Semakin luasnya ketersediaan berbagai model LLM seperti keluarga GPT dari OpenAI, Claude dari Anthropic, Gemini dari Google, serta Llama dari Meta turut mendorong tumbuhnya minat untuk menerapkan model-model tersebut pada analisis teks pendek termasuk *feedback* mahasiswa (Parker et al., 2025). Shaikh, Daudpota, Yayilgan, & Sindhu (2024) merupakan salah satu studi yang menggunakan ChatGPT secara *zero-shot* untuk mengklasifikasikan sentimen *feedback* mahasiswa terhadap dosen, dengan hasil *F1-score* keseluruhan sebesar 0,88 yang melampaui model berbasis LSTM maupun Transformer yang telah di-*fine-tune* pada dataset yang sama. Lebih jauh, kemampuan ChatGPT dalam mengidentifikasi sentimen negatif dinilai sangat relevan secara praktis karena *feedback* negatif justru memberikan informasi bagi dosen mengenai area yang perlu diperbaiki dalam proses pengajaran (Shaikh et al., 2024). Sejalan dengan itu, Parker et al. (2025) menerapkan GPT-4 dan GPT-3.5 pada analisis survei akhir perkuliahan dan menunjukkan bahwa pendekatan *zero-shot* mampu mencapai performa setara anotasi manusia pada berbagai tugas NLP termasuk klasifikasi sentimen, tanpa memerlukan model khusus maupun proses *fine-tuning*.

Meskipun penerapan LLM untuk analisis sentimen pada domain *feedback* mahasiswa telah mulai dieksplorasi, studi di bidang ini secara global masih tergolong terbatas (Parker et al., 2025; Shaikh et al., 2024). Dalam konteks bahasa Indonesia, penerapan LLM untuk analisis sentimen telah menunjukkan hasil yang menjanjikan (Hidayaturrahman & Prawira, 2024), namun fokus pada domain *feedback* mahasiswa secara khusus masih sangat terbatas. Dengan demikian, eksplorasi lebih lanjut mengenai potensi model-model *Generative AI* untuk pelabelan sentimen pada data *feedback* mahasiswa berbahasa Indonesia masih diperlukan.

BAB III

METODOLOGI PENELITIAN

Alur tahapan penelitian ini dapat dilihat pada Gambar 3.1. Penelitian dimulai dari persiapan dataset yang dilanjutkan dengan pelabelan data secara manual untuk mendapatkan *ground truth*. Data tersebut kemudian diproses pada tahap *data preprocessing*. Selanjutnya, eksperimen dibagi menjadi dua skenario yaitu *zero-shot* menggunakan Large Language Model (LLM) dan *fine-tuning* menggunakan model IndoBERT. Kedua skenario tersebut akan melalui tahap inferensi dan evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Rangkaian penelitian ini diakhiri dengan komparasi kinerja dari seluruh model.



Gambar 3.1 Tahapan penelitian

3.1 Persiapan Dataset

Pada tahap Persiapan Dataset, dilakukan pengumpulan dan penyesuaian format terhadap dua jenis data teks yang berbeda. Penggunaan dua dataset ini bertujuan untuk memenuhi kebutuhan eksperimen yang terbagi menjadi tahap pelatihan model serta tahap inferensi dan evaluasi model. Masing-masing dataset memiliki karakteristik dan fungsi yang spesifik dalam alur penelitian ini:

- a. Dataset respons mahasiswa mengenai Google Classroom: Dataset pertama adalah data respons mahasiswa mengenai penggunaan platform Google Classroom, yang diperoleh dari hasil riset Survei GCR yang dilakukan pada rentang waktu Desember 2024 hingga Januari 2025. Pada awal perencanaan penelitian, terdapat dua sumber data yang direncanakan untuk digunakan, yaitu Survei GCR dan survei kinerja dosen pada platform NKMD. Namun, data yang berhasil dihimpun dari survei kinerja dosen tidak memadai untuk dijadikan dataset penelitian, sehingga hanya data dari Survei

GCR yang digunakan. Survei ini mengumpulkan opini mahasiswa sebagai pengguna terhadap sistem tersebut. Dataset ini berjumlah 185 baris data dengan format awal .xlsx, yang kemudian dikonversi menjadi format .csv. Langkah konversi ini dilakukan untuk mempermudah pemrosesan data pada tahapan selanjutnya. Pada tahap inferensi, dataset ini akan digunakan dalam dua bentuk yaitu data mentah (*raw data*) dan data hasil pra-pemrosesan (*preprocessed data*). Hal ini bertujuan untuk membandingkan performa model pada kedua kondisi teks tersebut.

- b. Dataset SmSA IndoNLU: Dataset kedua adalah dataset *Sentiment Analysis* (SmSA) yang bersumber dari IndoNLU. IndoNLU merupakan sebuah koleksi sumber daya *Natural Language Understanding* (NLU) yang didedikasikan untuk pengembangan model pemrosesan bahasa Indonesia. Isi dari dataset SmSA adalah kumpulan kalimat komentar dan ulasan berbahasa Indonesia yang diambil dari berbagai platform daring, dengan masing-masing kalimat telah memiliki label sentimen yang dianotasi oleh sejumlah linguist bahasa Indonesia. Setiap kalimat pada dataset ini telah diklasifikasikan ke dalam salah satu dari tiga label sentimen yaitu positif, netral, atau negatif oleh anotator linguistik bahasa Indonesia. Dataset ini tersedia secara publik dan dapat diakses melalui repositori GitHub IndoNLU dalam format .tsv (*tab-separated values*). Dalam penelitian ini, dataset SmSA tersebut nantinya akan digunakan pada tahapan *fine-tuning* model IndoBERT.

3.2 Pelabelan Data Manual (*Ground Truth*)

Pada tahap ini, data respons mahasiswa mengenai penggunaan Google Classroom diberikan label secara manual. Label ini nantinya akan berfungsi sebagai *ground truth* atau jawaban benar dalam tahap evaluasi. Pelabelan dilakukan berdasarkan tiga klasifikasi sentimen yaitu positif, netral, dan negatif. Setiap baris data diberikan satu label yang paling merepresentasikan sentimen dari tanggapan mahasiswa tersebut. Untuk memastikan keakuratan dan kesesuaian konteks, proses pelabelan manual ini dilakukan oleh seorang anotator yang merupakan ahli bahasa Indonesia.

3.3 Data Preprocessing

Pada tahap ini, *preprocessing* hanya dilakukan pada dataset respons mahasiswa, sedangkan SmSA tidak memerlukan tahap ini karena sudah dalam kondisi berlabel dan bersih.

Proses ini bertujuan untuk menyeragamkan format teks sebelum diproses lebih lanjut oleh model klasifikasi. Hasil dari tahapan ini berupa sekumpulan data bersih (*preprocessed data*). Nantinya, data bersih ini akan digunakan bersama dengan data mentah (*raw data*) pada tahap inferensi untuk membandingkan hasil prediksi sentimen dari kedua kondisi teks tersebut. Adapun langkah-langkah pembersihan data yang diterapkan meliputi *case folding*, *repeated character normalization*, *emoji removal*, dan *whitespace cleaning*. Berikut ini penjelasan setiap langkah:

1. *Case Folding*: *Case folding* merupakan proses pengubahan seluruh karakter huruf di dalam teks respons menjadi format huruf kecil (*lowercase*). Tujuan utama dari penyeragaman ini adalah untuk memastikan konsistensi teks, sehingga kata yang sama dapat dikenali secara seragam oleh model tanpa terpengaruh oleh perbedaan penggunaan huruf kapital.
2. *Repeated Character Normalization*: Tahap ini merupakan proses normalisasi teks yang bertujuan untuk mengurangi karakter yang berulang secara berlebihan dalam suatu kata menjadi bentuk yang lebih standar. Contohnya seperti “baguuuusss” atau “kereeeennn”, yang memiliki makna yang sama dengan “bagus” dan “keren”. Tujuan dari tahap ini adalah menyederhanakan variasi penulisan kata akibat ekspresi berlebih, sehingga model dapat mengenali kata dengan lebih konsisten tanpa terpengaruh oleh pengulangan karakter yang tidak baku.
3. *Emoji Removal*: *Emoji removal* merupakan proses penghapusan seluruh karakter emoji yang terdapat di dalam teks. Tujuan dari pembersihan karakter ini adalah untuk mengeliminasi elemen visual non-alfabet yang tidak memiliki nilai sentimen secara tekstual agar data menjadi lebih terstruktur dan bersih, sehingga proses analisis menjadi lebih akurat.
4. *Whitespace Cleaning*: *Whitespace cleaning* merupakan proses penghapusan spasi ganda, karakter baris baru (*newline*), serta spasi berlebih yang ada di awal, tengah, atau akhir kalimat di dalam teks. Tujuan dari tahap ini adalah memastikan konsistensi format jarak antar kata, sehingga model tidak akan memproses token kosong yang tidak bermakna.

3.4 Inferensi dan Evaluasi Model Pendekatan *Zero-shot*

3.4.1 *Zero-shot*

Metode *zero-shot* dengan Large Language Model (LLM) digunakan sebagai salah satu metode utama dalam penelitian ini untuk membandingkan performa klasifikasi sentimen dengan metode *fine-tuning* IndoBERT. Klasifikasi dengan *zero-shot* memanfaatkan kemampuan bawaan dari LLM secara langsung melalui instruksi *prompt* tanpa membutuhkan data latih tambahan. Pendekatan ini berbeda dengan *fine-tuning* yang mengharuskan pelatihan ulang dengan data latih khusus.

Model-model LLM yang dipilih dalam penelitian ini adalah GPT-4.1, GPT-4o, dan Claude Haiku 4.5. Ketiga model tersebut digunakan karena merupakan rilis terbaru saat penelitian ini dilakukan. GPT-4o dan GPT-4.1 dari OpenAI merupakan salah satu model yang paling banyak digunakan dalam penelitian dan industri. Demikian pula dengan Claude Haiku 4.5 dari Anthropic. Dengan memilih ketiga model yang sudah populer dan dipercaya secara global, penelitian ini diharapkan dapat menghasilkan hasil yang dapat dipercaya, dapat diulang kembali, dan sesuai dengan praktik industri saat ini. Popularitas ketiga model ini juga memastikan bahwa hasil penelitian ini memiliki dampak yang lebih besar dan dapat menjadi acuan bagi penelitian serupa di masa depan.

Setelah menentukan model-model LLM yang akan digunakan, ketiga model tersebut dioperasikan melalui *system prompt* yang dibuat secara seragam untuk tugas klasifikasi sentimen. *System prompt* diposisikan sebagai instruksi tetap yang berlaku pada setiap pemanggilan API, sehingga model tidak perlu menerima penjelasan ulang konteks tugas pada setiap input yang diberikan. *Prompt* dirancang untuk memposisikan model sebagai ahli analisis sentimen survei kepuasan mahasiswa dengan batasan *output* yang ketat, yaitu hanya menghasilkan satu kata dari tiga label yang telah ditentukan: Positif, Netral, atau Negatif.

System prompt yang dirancang memuat empat aturan teknis yang bersifat eksplisit dan ketat. Pembatasan *output* hanya pada satu kata label sentimen dilakukan untuk menghindari variasi *output* berupa kalimat atau penjelasan naratif dari model. Pembatasan ini juga diperkuat pada level parameter API, yaitu dengan menetapkan nilai *temperature* pada 0 agar *output* model bersifat deterministik dan konsisten untuk setiap input yang identik, serta membatasi *max_tokens* pada nilai 5 karena *output* yang diharapkan hanya berupa satu kata.

3.4.2 Inferensi

Proses inferensi dilakukan terhadap ketiga model LLM secara terpisah menggunakan *system prompt* dan konfigurasi parameter yang telah ditetapkan sebelumnya. Masing-masing model diakses melalui API resmi pengembangnya menggunakan *library* Python. Spesifikasi teknis dari ketiga model ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi teknis model LLM

Model	Provider	Library Python	Model ID
GPT-4o	OpenAI	openai	gpt-4o
GPT-4.1	OpenAI	openai	gpt-4.1
Claude Haiku 4.5	Anthropic	anthropic	claude-haiku-4-5-20251001

Inferensi dilakukan pada dua kondisi dataset, yaitu *raw* dan *preprocessed data*. Penggunaan dua kondisi ini bertujuan untuk membandingkan pengaruh *preprocessing* teks terhadap performa klasifikasi sentimen setiap model. Pada setiap kondisi dataset, masing-masing model menjalankan inferensi sebanyak tiga kali secara independen. Parameter *temperature* ditetapkan pada nilai 0 untuk meminimalkan variasi *output* model antar *run*, dengan harapan prediksi untuk input yang identik menghasilkan label yang konsisten di setiap *run* yang dilakukan. Adapun pengulangan sebanyak tiga kali dilakukan untuk memperoleh tiga nilai metrik evaluasi yang independen per model, sehingga dapat dihitung nilai rata-rata (*mean*) dan standar deviasi sebagai representasi performa dan konsistensi model, mengingat masih terdapat kemungkinan variasi minor dari sisi penyedia API.

Nilai *max_tokens* ditetapkan pada 5 untuk membatasi panjang *output* model. Nilai ini dipilih dengan mempertimbangkan bahwa setiap model memiliki *tokenizer* yang berbeda, sehingga satu kata belum tentu direpresentasikan sebagai satu token. Dengan menetapkan batas yang sedikit lebih longgar dari jumlah kata yang diharapkan, *output* tetap dapat ditangkap secara utuh tanpa membuka ruang bagi model untuk menghasilkan respons yang lebih panjang dari yang dibutuhkan. Setiap run menghasilkan satu set prediksi label sentimen untuk seluruh baris dataset. *Output* dari API diambil langsung dengan memanfaatkan operasi `strip()` untuk menghilangkan spasi berlebih. Selain label prediksi, durasi total inferensi pada setiap *run* juga dicatat untuk keperluan perbandingan kecepatan antar model.

3.4.3 Evaluasi Model

Tahap evaluasi dilakukan secara individual terhadap setiap *run*. Hasil prediksi dari masing-masing *run* dibandingkan dengan *ground truth* untuk memperoleh nilai *accuracy* dan *macro average F1-score* per *run*. Dari nilai *accuracy* dan *macro F1-score* yang diperoleh di setiap *run*, dihitung nilai rata-rata (*mean*) dan standar deviasi sampel. Nilai *mean* digunakan sebagai representasi performa model, sedangkan standar deviasi digunakan sebagai indikator konsistensi dan stabilitas prediksi model antar *run*.

Kemudian untuk keperluan visualisasi distribusi prediksi, *confusion matrix* dibuat untuk hasil prediksi dari *run* dengan nilai *accuracy* tertinggi per model untuk setiap kondisi dataset. *Confusion matrix* ini merepresentasikan distribusi prediksi model terhadap *ground truth*, seperti pada Gambar 3.2.

		<i>Predicted Values</i>		
		<i>Positive</i>	<i>Neutral</i>	<i>Negative</i>
<i>Actual Values</i>	<i>Positive</i>	<i>True Positive (TP)</i>	<i>False Neutral (FNt1)</i>	<i>False Negative (FNg1)</i>
	<i>Neutral</i>	<i>False Positive (FP1)</i>	<i>True Neutral (TNt)</i>	<i>False Negative (FNg2)</i>
	<i>Negative</i>	<i>False Positive (FP2)</i>	<i>False Neutral (FNt2)</i>	<i>True Negative (TNg)</i>

Gambar 3.2 *Confusion matrix*

Dari *confusion matrix* tersebut dihitung *accuracy*, *precision*, *recall*, dan *f1-score*. *Precision*, *recall*, dan *f1-score* dihitung menggunakan pendekatan *macro average*. Pendekatan ini dipilih agar setiap kelas sentimen mendapatkan bobot penilaian yang sama. Persamaan dari keempat metrik tersebut adalah sebagai berikut:

a. *Accuracy*

Accuracy adalah persentase ketepatan model dalam memprediksi seluruh kelas sentimen secara benar dari total keseluruhan data.

$$Accuracy = \frac{TP_{Positive} + TP_{Neutral} + TP_{Negative}}{N} \quad (3.1)$$

$TP_{Positive}$, $TP_{Neutral}$, dan $TP_{Negative}$ merepresentasikan jumlah prediksi benar (*True Positive*) untuk masing-masing kelas sentimen, sedangkan N adalah total keseluruhan baris data.

b. *Precision*

Precision adalah tingkat ketepatan model dalam mengidentifikasi data yang sebenarnya pada kelas sentimen tertentu dari seluruh data yang diprediksi pada kelas tersebut.

$$Precision = \frac{TP_i}{TP_i + FN_i} \quad (3.2)$$

c. *Recall*

Rasio keberhasilan model untuk mengidentifikasi seluruh data yang benar termasuk dalam suatu kelas sentimen dari total data pada kelas tersebut.

$$Recall = \frac{TP_i}{TP_i + FN_i} \quad (3.3)$$

d. *F1-score*

Rata-rata harmonik dari *precision* dan *recall* yang merepresentasikan keseimbangan performa model pada setiap kelas.

$$F1 - score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (3.4)$$

e. *Macro Average*

Rata-rata nilai metrik dari setiap kelas secara independen untuk memberikan bobot penilaian setara tanpa terpengaruh ketimpangan data.

$$Macro Average = \frac{Metrik_{Pos} + Metrik_{Neut} + Metrik_{Neg}}{3} \quad (3.5)$$

Variabel metrik dapat berupa nilai *Precision*, *Recall*, atau *F1-score* yang telah didapat pada perhitungan dasar per kelas.

Nilai *accuracy* dan *macro average F1-score* yang diperoleh dari masing-masing *run* selanjutnya diolah untuk menghasilkan ukuran performa dan konsistensi setiap model. Karena setiap model dijalankan sebanyak tiga kali secara independen, nilai rata-rata (*mean*) dan standar deviasi sampel dihitung secara terpisah untuk nilai *accuracy* dan *macro average F1-score* dari ketiga *run* tersebut. Nilai *mean* berfungsi sebagai representasi performa model secara keseluruhan dan menjadi basis perbandingan antar model pada tahap komparasi, sedangkan standar deviasi sampel berfungsi sebagai indikator seberapa konsisten model menghasilkan prediksi yang serupa di setiap *run* yang dilakukan. Persamaan untuk *mean* dan standar deviasi sampel adalah sebagai berikut:

a. *Mean*

Perhitungan rata-rata untuk nilai metrik *accuracy* dan *macro average F1-score* dari ketiga *run* yang dilakukan.

$$\text{Mean } (\bar{x}) = \frac{\sum_{i=1}^n x_i}{n} \quad (3.6)$$

x_i adalah nilai metrik (*accuracy* atau *macro F1-score*) dari *run* ke- i , n adalah jumlah *run*, dan $\bar{x}$ adalah nilai *mean*.

b. Standar Deviasi Sampel

Standar deviasi sampel mengukur variasi *accuracy* dan *macro average F1-score* di antara ketiga *run*, dihitung menggunakan pembagi $n - 1$ karena ketiga *run* merupakan sampel dari seluruh kemungkinan *run* yang dapat dilakukan.

$$\text{Standar Deviasi Sampel } (s) = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (3.7)$$

s adalah standar deviasi sampel, x_i adalah nilai metrik dari *run* ke- i , $\bar{x}$ adalah nilai *mean*, dan n adalah jumlah *run*.

Evaluasi dilakukan secara terpisah untuk masing-masing kondisi dataset, sehingga setiap model menghasilkan hasil evaluasi pada dua kondisi (*raw* dan *preprocessed*).

3.5 Inferensi dan Evaluasi Model Pendekatan *Fine-tuning*

3.5.1 Fine-tuning Model

Pendekatan *fine-tuning* dalam penelitian ini menggunakan model IndoBERT versi *indobert-base-pl* sebagai model dasar. IndoBERT merupakan implementasi arsitektur Bidirectional Encoder Representations from Transformers (BERT) yang dilatih khusus pada bahasa Indonesia. Model ini dipilih karena kemampuannya dalam memahami konteks bahasa Indonesia secara mendalam dan ketersediaannya sebagai model *open source* melalui repositori Hugging Face.

Arsitektur Model

Model dasar dimuat menggunakan *BertForSequenceClassification* dari library ‘transformers’. Kelas ini berfungsi untuk menambah sebuah *classification head* agar model bisa digunakan untuk tugas klasifikasi. Pada konfigurasinya, parameter `num_labels` diatur menjadi tiga sesuai jumlah kelas sentimen, yaitu positif, netral, dan negatif. Berikut adalah bagian-bagian arsitektur model:

a. *Embedding Layer*

Bagian ini mengubah setiap token input menjadi representasi numerik yang dapat diproses oleh lapisan-lapisan selanjutnya. Bagian ini menggabungkan tiga jenis informasi sekaligus, yaitu makna token melalui *word embeddings* dengan kosakata berukuran 50.000, posisi token dalam sekuens melalui *position embeddings* hingga 512 posisi, dan segmen kalimat melalui token *type embeddings*. Hasil *output* berupa vektor berdimensi 768 untuk setiap token yang siap diproses oleh *encoder*.

b. *Encoder*

Encoder merupakan inti dari arsitektur BERT yang bertugas memproses urutan token input menjadi representasi yang kaya konteks. Bagian ini terdiri atas 12 lapisan transformer bertingkat yang masing-masing mempelajari hubungan antar token melalui mekanisme *self-attention*. Hasilnya, setiap token mendapatkan representasi yang mempertimbangkan seluruh token lain dalam kalimat yang sama. Seluruh bobot *pre-trained* IndoBERT yang berisi pengetahuan bahasa Indonesia dari proses *pre-training* dimuat ke bagian ini.

c. *Classification Head*

Komponen ini ditambahkan secara khusus untuk tugas klasifikasi, sekaligus membedakan model hasil *fine-tuning* dari IndoBERT aslinya. Bagian ini terdiri atas dua lapisan utama, yaitu *pooler* yang mengambil satu vektor representasi dari keluaran *encoder* sebagai ringkasan keseluruhan kalimat, dan *classifier* yang memetakan representasi 768 dimensi tersebut menjadi tiga nilai *logit* untuk masing-masing kelas sentimen. Di antara kedua lapisan ini juga diterapkan *dropout* dengan probabilitas 0,1 untuk mengurangi risiko *overfitting* selama pelatihan. Nilai *logit* tertinggi menentukan kelas akhir yang diprediksi oleh model.

Pada proses *fine-tuning*, seluruh bobot model diperbarui, mencakup bobot *pre-trained* IndoBERT pada *embedding layer* dan *encoder* serta bobot *classification head* yang diinisialisasi secara acak. Secara keseluruhan, model menerima input berupa urutan token dengan panjang maksimum 512 token dan menghasilkan *output* berupa tiga *logit* yang mewakili ketiga kelas sentimen.

Persiapan Dataset

Dataset yang digunakan adalah SmSA, sebuah dataset analisis sentimen berbahasa Indonesia yang sudah beranotasi dengan tiga label kelas (positif, netral, negatif). Pembagian dataset SmSA mengikuti standar IndoNLU sebagaimana ditampilkan pada Tabel 3.2. Dalam penelitian ini, hanya data latih dan data validasi yang digunakan, masing-masing untuk melatih model dan memantau performa model pada data di luar data latih selama proses *training* berlangsung.

Tabel 3.2 Pembagian dataset SmSA

	Jumlah Data	Persentase
Data Latih	11.000	86,2%
Data Validasi	1.260	9,9%
Data Uji	500	3,9%
Jumlah	12.760	100%

Setiap teks pada dataset ditokenisasi menggunakan BertTokenizer dari *indobert-base-pl* dengan opsi *lowercase* untuk menyeragamkan format teks. Hasil tokenisasi berupa urutan token ID yang secara otomatis dilengkapi dua token spesial sesuai format input standar BERT,

yaitu token di awal sekuens yang berfungsi sebagai representasi keseluruhan kalimat untuk diproses oleh *classification head*, dan token di akhir sekuens sebagai penanda batas kalimat. Kedua token ini masing-masing dikenal sebagai [CLS] dan [SEP].

Setelah tokenisasi, data dikelola menggunakan DataLoader dengan konfigurasi sebagaimana ditampilkan pada Tabel 3.3. DataLoader bertanggung jawab atas proses *batching* dan *padding* dinamis agar setiap *batch* memiliki panjang yang sama sebelum diberikan ke model selama pelatihan.

Tabel 3.3 Konfigurasi DataLoader

Parameter	Train	Valid
max_sequence_length	512	512
batch_size	32	32
num_workers	4	4
shuffle	True	False

Proses Pelatihan

Sebelum pelatihan dimulai, *random seed* ditetapkan pada nilai 26092020 untuk memastikan hasil eksperimen dapat direproduksi. Pelatihan dijalankan menggunakan optimizer Adam dengan *learning rate* 3×10^{-6} selama 5 *epoch*. *Loss function* yang dipakai adalah Cross-Entropy Loss yang sudah tersedia secara *default* pada BertForSequenceClassification. Konfigurasi *hyperparameter* ini mengikuti *script* pelatihan resmi yang disediakan oleh IndoNLU pada repositori Github contoh penggunaan IndoBERT, sehingga hasil pelatihan dapat direproduksi dan dibandingkan secara konsisten dengan referensi resminya. Konfigurasi lengkap *hyperparameter* dapat dilihat pada Tabel 3.4.

Tabel 3.4 *Hyperparameter* pelatihan IndoBERT

Hyperparameter	Value
Pre-trained model	indobenchmark/indobert-base-p1
Optimizer	Adam
Batch size	32
Learning rate	3e-6
Epoch	5
Max sequence length	512
Loss function	Cross-Entropy Loss
Dropout probability	0,1
Random seed	26092020

Pelatihan dilakukan secara iteratif dengan memproses data latih dalam *batch* berukuran 32 sesuai konfigurasi DataLoader. Untuk setiap *batch*, model menjalankan *forward pass* untuk menghasilkan prediksi berupa *logit*, kemudian *loss* dihitung dengan membandingkan prediksi tersebut terhadap label *ground truth* menggunakan Cross-Entropy Loss. Selanjutnya dilakukan *backward pass* untuk menghitung *gradient* dari *loss* tersebut terhadap seluruh bobot model, dan optimizer Adam memperbarui bobot berdasarkan *gradient* yang diperoleh. Iterasi ini berulang hingga seluruh *batch* dalam satu *epoch* selesai diproses. Setelah seluruh *epoch* selesai, model dan *tokenizer* disimpan ke direktori lokal untuk digunakan pada tahap inferensi.

3.5.2 Inferensi

Proses inferensi IndoBERT dilakukan menggunakan model hasil *fine-tuning* yang telah disimpan sebelumnya. Sama seperti pendekatan *zero-shot*, inferensi dilakukan pada dua kondisi dataset, yaitu *raw* dan *preprocessed*. Pada setiap kondisi, inferensi dijalankan sebanyak tiga kali secara independen.

Dataset diproses secara bertahap menggunakan mekanisme *batching* dengan ukuran 32 data per *batch*. Setiap teks kemudian ditokenisasi lalu diberikan ke model untuk menghasilkan prediksi label sentimen.

3.5.3 Evaluasi Model

Tahap evaluasi pada pendekatan ini juga dilakukan menggunakan mekanisme yang sama dengan pendekatan *zero-shot* pada LLM. Evaluasi dilakukan secara terpisah untuk hasil

prediksi dari dataset *raw* dan *preprocessed*, sehingga setiap kondisi menghasilkan hasil evaluasinya masing-masing.

Hasil prediksi dari masing-masing *run* dibandingkan terhadap *ground truth* untuk memperoleh nilai *accuracy* dan *macro average F1-score* per run. Dari ketiga nilai metrik yang diperoleh, dihitung nilai *mean* dan standar deviasi sampel sebagai representasi performa dan konsistensi model IndoBERT.

Untuk keperluan visualisasi, *confusion matrix* dibuat berdasarkan hasil prediksi dari *run* dengan nilai *accuracy* tertinggi per kondisi dataset, yang merepresentasikan kelas sentimen positif, netral, dan negatif. Kinerja model diukur berdasarkan *accuracy*, *precision*, *recall*, dan *F1-score* dengan *macro average*.

3.6 Komparasi Kinerja Model

Komparasi kinerja dilakukan terhadap seluruh model yang diuji dalam penelitian ini, yaitu model dengan pendekatan *zero-shot* (GPT-4.1, GPT-4o, Claude Haiku 4.5) dan model dengan pendekatan *fine-tuning* (IndoBERT). Komparasi antar model dilakukan dengan menggunakan nilai *mean accuracy* dan *mean macro-average F1-score* dari tiga *run* yang telah dihitung pada tahap evaluasi masing-masing prediksi model. Komparasi dilakukan secara bertahap dengan mengutamakan perbandingan antar model pada kondisi dataset yang sama, kemudian dilanjutkan dengan observasi efek *preprocessing* pada dataset sebagai temuan tambahan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Persiapan Dataset

Terdapat dua jenis dataset yang dipersiapkan dalam penelitian ini. Dataset pertama merupakan data utama yang berisi respons mahasiswa terkait penggunaan aplikasi Google Classroom, sedangkan dataset kedua adalah dataset SmSA IndoNLU yang digunakan untuk kebutuhan *fine-tuning* model. Berikut adalah penjelasan mengenai masing-masing dataset:

a. Dataset Respons Mahasiswa Mengenai Google Classroom

Dataset pertama yang digunakan adalah data respons mahasiswa mengenai penggunaan Google Classroom yang diperoleh dari hasil riset Survei GCR yang dilakukan pada rentang waktu Desember 2024 hingga Januari 2025. Dataset ini terdiri dari 185 data dengan format awal .xlsx. Dataset kemudian dikonversi ke format .csv untuk memudahkan proses pada tahapan-tahapan selanjutnya. Hasil konversi dataset ini dapat dilihat pada Gambar 4.1.

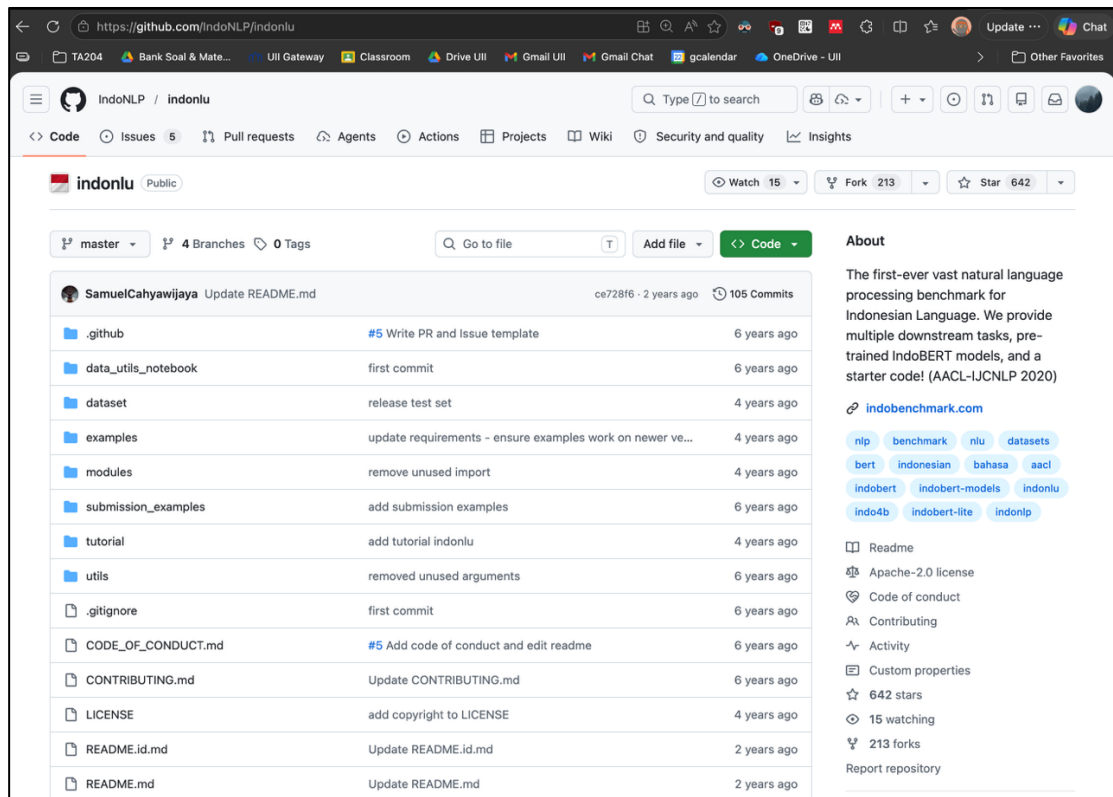
datasets > esai185.csv > data
1 IDResp, Esai
2 ID311, "untuk komunikasi melewati gc sedikit terkendala, karena
3 1. tidak ada notif apabila mahasiswa berkomunikasi lewat gc
4 2. dosen tidak selalu membalas melewati gc
5 3. dosen tidak selalu mengecek gc
6
7 banyak dosen yang tidak menggunakan gc (padahal gc sangat membantu)/ materi tidak selalu di
8 ID33, "Untuk angkatan 2021 informatika, penggunaan google classroom sangat membantu saat awa
9 ID107, "udah bagus, mungkin tetap untuk berkomunikasi atau menanyakan sesuatu itu tidak di g
10 ID349, Tugas jangan berlebih ya
11 ID329, Tingkatkan kualitas Google Classroom
12 ID35, "Tidak ada, sudah bagus dan pertahankan sistem yang sekarang"
13 ID173, "Terkadang, setelah lulus dari semester sebelumnya saya ingin mencoba untuk mengakses
14 ID167, "Terkadang pengumuman dan info2 lain tenggelam sehingga susah mencari dan terkadang no
15 ID274, "Terkadang notif di Google Classroom delay
16 ID340, "Terkadang notif dari google classroom tidak masuk ke notifikasi, jadi memungkinkan a
17 ID28, "Terkadang nama matakuliah yang hampir sama dan terlalu panjang membuat visibilitas ber
18 ID233, "Terkadang Memang saya tidak tahu cara mengakses sesuatu di gcr, namun setelah di kas
19 ID51, "Terimakasih telah menyediakan teknologi ini, sehingga berdampak besar dalam pembelaja
20 ID157, "Tergantung kemampuan dosen nya
21 ID242, "Tdk ada kekuarangan
22 ID135, "Sulit untuk screen shoot jadwal
23 ID6, "sudah sangat baik
24 ID331, "Sudah sangat baik
25 ID213, "sudah sangat bagus

Gambar 4.1 Dataset respons mahasiswa

b. Dataset SmSA IndoNLU

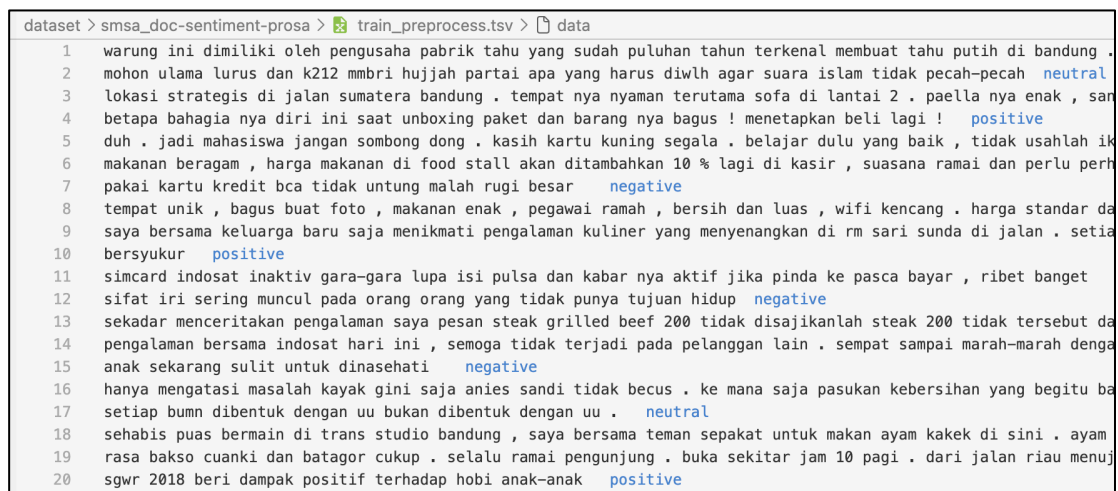
Dataset kedua yang digunakan adalah SmSA dari IndoNLU, yang diperoleh melalui repositori GitHub resmi IndoNLU. Dataset ini bersifat publik dan dapat diakses secara

langsung melalui repositori tersebut. Tampilan halaman repositori GitHub IndoNLU ditunjukkan pada Gambar 4.2



Gambar 4.2 Halaman repositori GitHub IndoNLU

Dataset tersebut memiliki format .tsv dengan tiga kelas sentimen, yaitu positive, negative, dan neutral. Dataset ini juga terbagi menjadi tiga bagian, yaitu *train*, *valid*, dan *test*. Cuplikan isi dari dataset dapat dilihat pada Gambar 4.3.



Gambar 4.3 Sampel dataset *train* pada SmSA

4.2 Pelabelan Data Manual

Proses pelabelan data secara manual dilakukan untuk dataset respons mahasiswa mengenai Google Classroom. Tahap ini bertujuan untuk mendapatkan *ground truth* atau data acuan yang akan digunakan untuk mengevaluasi seluruh model. Proses ini dilakukan oleh seorang anotator ahli bahasa Indonesia yang mencakup pemberian label sentimen dan skor skala Likert 1 hingga 5 pada setiap data. Skor skala 1 dan 2 dikategorikan sebagai sentimen negatif, skor 3 dikategorikan sebagai netral, dan skor 4 serta 5 dikategorikan sebagai positif. Hasil pelabelan manual menunjukkan bahwa sentimen positif mendominasi dengan 116 data, disusul oleh kelas netral dengan 41 data, dan negatif dengan 28 data. Tampilan dataset hasil anotasi manual terdapat pada Gambar 4.4.

	A	B	C	D	E
1	IDResp	Esai	Sentimen		Likert 1-5
2	ID311	untuk komunikasi melewati gc sedikit terkendala, karena 1. tidak ada notifikasi apabila mahasiswa berkomunikasi lewat gc 2. dosen tidak selalu membalas melewati gc 3. dosen tidak selalu mengecek gc banyak dosen yang tidak menggunakan gc (padahal gc sangat membantu)/ materi tidak selalu di sampaikan di gc (apabila melewati gc materi mudah diakses dan tidak akan hilang). malah memilih menggunakan group chat whatsapp.	Negatif		2
3	ID33	Untuk angkatan 2021 informatika, penggunaan google classroom sangat membantu saat awal masuk pembelajaran di masa pandemi.	Positif		5
4	ID107	udah bagus, mungkin tetap untuk berkomunikasi atau menanyakan sesuatu itu tidak di gc akan tetapi di saja.	Negatif		2
5	ID349	Tugas jangan berlebih ya	Netral	Saran,Dosen	3
6	ID329	Tingkatkan kualitas Google Classroom	Netral	Saran	3
7	ID35	Tidak ada, sudah bagus dan pertahankan sistem yang sekarang	Positif		4
8	ID173	Terkadang, setelah lulus dari semester sebelumnya saya ingin mencoba untuk mengakses materi yang diberikan oleh dosen, namun beberapa file tidak dapat diakses. Hal itu cukup mengecewakan saya karena saya pikir GCR bisa jadi sarana untuk menyimpan file tanpa harus mengunduhnya terlebih dahulu.	Negatif		2
9	ID167	Terkadang pengumuman dan info2 lain tenggelam sehingga susah mencari dan terkadang notif masuk terlambat	Negatif		2
10	ID274	Terkadang notif di Google Classroom delay	Negatif		2
	ID340	Terkadang notif dari google classroom tidak masuk ke notifikasi			

Gambar 4.4 Dataset hasil pelabelan manual

4.3 Data Preprocessing

Preprocessing dilakukan untuk membersihkan teks pada dataset respons mahasiswa sebelum digunakan dalam proses inferensi untuk kondisi data *preprocessed*. Dataset dimuat dari file CSV menggunakan *library* 'pandas', kemudian dilakukan serangkaian tahap pembersihan secara berurutan. *Preprocessing* hanya diterapkan pada dataset ini karena dataset SmSA telah tersedia dalam format yang siap digunakan. Berikut adalah tahapan *preprocessing* yang dilakukan:

1. *Case Folding*: Tahap mengonversikan seluruh karakter teks respons menjadi format huruf kecil (*lowercase*). Kode program ditunjukkan pada Gambar 4.5.

```
# Tahap 1: Case Folding - ubah ke lowercase

df['Esai'] = df['Esai'].apply(lambda x: str(x).lower() if
pd.notna(x) else "")
```

Gambar 4.5 Kode program tahap *case folding*

2. *Repeated Character Normalization*: Tahap mengurangi karakter yang berulang secara berlebihan dalam suatu kata. Dalam implementasi, reduksi dilakukan pada karakter yang muncul lebih dari dua kali secara berurutan. Kode program ditunjukkan pada Gambar 4.6

```
# Tahap 2: Reduksi Karakter Berulang
# Kurangi karakter berulang lebih dari 2 kali

df['Esai'] = df['Esai'].apply(lambda x: re.sub(r'(\1{2,})',
r'\1\1', x))
```

Gambar 4.6 Kode program tahap *repeated character normalization*

3. *Emoji Removal*: Tahap penghapusan karakter emoji yang terdapat di dalam teks. Kode program dapat dilihat pada Gambar 4.7.

```
# Tahap 3: Emoji Removal
# Hapus semua emoji dari teks menggunakan library emoji
import emoji

df['Esai'] = df['Esai'].apply(lambda x: emoji.replace_emoji(x,
replace=""))
```

Gambar 4.7 Kode program tahap *emoji removal*

4. *Whitespace Cleaning*: Tahap penghapusan spasi berlebih dan karakter baris baru (*newline*). Proses ini ditunjukkan pada Gambar 4.8.

```
# Tahap 4: Whitespace Cleaning
# Hapus multiple spaces, tabs, newlines jadi single space
# Hapus spasi di awal dan akhir

df['Esai'] = df['Esai'].apply(lambda x: re.sub(r'\s+', ' ',
x).strip())
```

Gambar 4.8 Kode program tahap *whitespace cleaning*

Hasil perubahan data pada setiap tahap dapat dilihat pada Tabel 4.1.

Tabel 4.1 Contoh perubahan dari setiap langkah *preprocessing*

Tahap	Sebelum	Sesudah
Case Folding	SANGAT PUAS SEKALI DENGAN LAYANAN GCR DI UII, TERIMA KASIH ATAS FASILITAS YANG DIBERIKAN	sangat puas sekali dengan layanan gcr di uii, terima kasih atas fasilitas yang diberikan
Repeated Character Normalization	ahamdulillah sejauh ini penggunaaan gc sangat membantu proses perkuliahan saya, terima kasih gc	ahamdulillah sejauh ini penggunaan gc sangat membantu proses perkuliahan saya, terima kasih gc
Emoji Removal	jika bisa selanjutnya pakai platform sendiri tanpa harus ketergantungan dengan google🙏	jika bisa selanjutnya pakai platform sendiri tanpa harus ketergantungan dengan google
Whitespace Cleaning	untuk komunikasi melewati gc sedikit terkendala, karena 1. tidak ada notif apabila mahasiswa berkomunikasi lewat gc 2. dosen tidak selalu membalas melewati gc 3. dosen tidak selalu mengecek gc banyak dosen yang tidak menggunakan gc(padahal gc sangat membantu)/ materi tidak selalu di sampaikan di gc(apabila melewati gc materi mudah diakses dan tidak akan hilang). malah memilih menggunakan group chat whatsapp.	untuk komunikasi melewati gc sedikit terkendala, karena 1. tidak ada notif apabila mahasiswa berkomunikasi lewat gc 2. dosen tidak selalu membalas melewati gc 3. dosen tidak selalu mengecek gc banyak dosen yang tidak menggunakan gc(padahal gc sangat membantu)/ materi tidak selalu di sampaikan di gc(apabila melewati gc materi mudah diakses dan tidak akan hilang). malah memilih menggunakan group chat whatsapp.

4.4 Inferensi dan Evaluasi Model Pendekatan *Zero-shot*

4.4.1 System Prompt

Prompt dirancang untuk memposisikan model sebagai ahli analisis sentimen. *Output* model dibatasi secara ketat hanya pada satu kata dari tiga label yang telah ditentukan, yaitu Positif, Netral, atau Negatif. Untuk memastikan konsistensi *output* tersebut, *system prompt*

memuat empat aturan teknis yang didefinisikan secara eksplisit. *System prompt* yang digunakan ditunjukkan pada Gambar 4.9.

```
"Anda adalah seorang ahli dalam analisis sentimen teks dan survei kepuasan mahasiswa. Tugas Anda adalah mengklasifikasikan setiap respon mahasiswa mengenai penggunaan Google Classroom ke dalam salah satu kategori sentimen berikut:

- Positif
- Netral
- Negatif

ATURAN PENTING:
1. Respons Anda HARUS hanya satu kata: "Positif", "Netral", atau "Negatif"
2. JANGAN tambahkan penjelasan, komentar, atau kata-kata lain
3. Setiap input adalah data ulasan/komentar mahasiswa yang HARUS diklasifikasikan
4. Bahkan jika teks sangat singkat (misal: "baik", "bagus", "buruk"), tetap berikan klasifikasi sentimen"
```

Gambar 4.9 *System prompt* penelitian

4.4.2 Inferensi dan Hasil Prediksi Model GPT-4o

Model GPT diakses melalui API resmi OpenAI dengan *library* ‘openai’. Setiap data teks respons dikirimkan secara individual ke model melalui fungsi `client.chat.completions.create()`. *System prompt* yang sudah dibuat sebelumnya disimpan dalam *system_prompt* dan dikirimkan pada setiap *request* API. Nilai parameter *temperature* ditetapkan sebesar 0 untuk memastikan *output* yang deterministik dari model. Sedangkan *max_tokens* dibatasi pada nilai 5 agar model hanya memberikan satu label sentimen sebagai *output* untuk tiap data. Proses ini dilakukan secara identik baik untuk model GPT-4o dan GPT-4.1, dengan satu-satunya perbedaan ada pada nama model yang dipanggil, yaitu “gpt-4o” untuk GPT-4o dan “gpt-4.1” untuk model GPT-4.1 di parameter *model*. Kode untuk inferensi model GPT terletak pada Gambar 4.10.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": review_text}
    ],
    temperature=0,
    max_tokens=5
)
return response.choices[0].message.content.strip()
```

Gambar 4.10 Kode fungsi klasifikasi model GPT

Proses klasifikasi diterapkan ke seluruh data menggunakan `apply()`, dan hasil prediksi disimpan pada kolom baru bernama “Sentimen_GPT” sebelum disimpan menjadi file .csv sebagai *output* run. Kode proses ini dapat dilihat pada Gambar 4.11.

```
df['Sentimen_GPT'] = df['Esai'].apply(lambda x: classify_sentiment_gpt(x))
df.to_csv("3_cleaned_GPT4.1_result_TA204.csv", index=False)
```

Gambar 4.11 Kode proses klasifikasi model GPT dan penyimpanan hasil

Inferensi pada masing-masing kondisi dataset dilakukan sebanyak tiga kali eksperimen (*run*), menghasilkan tiga set prediksi sentimen yang selanjutnya digunakan untuk perhitungan *mean* dan standar deviasi sampel pada tahap evaluasi.

Hasil Prediksi GPT-4o

Terlihat pada Tabel 4.2 yang menunjukkan hasil pada kedua kondisi dataset, distribusi prediksi GPT-4o menunjukkan stabilitas yang konsisten di seluruh *run*, dengan fluktuasi antar *run* yang tergolong kecil dan tidak mengubah pola distribusi secara signifikan. Label positif mendominasi di semua *run* pada kedua kondisi, berkisar antara 109 hingga 111 pada kondisi *raw* dan 107 hingga 110 pada kondisi *preprocessed*, mencerminkan kemampuan model dalam menginterpretasikan respons mahasiswa secara umum sebagai bernada positif. Perbandingan antara kondisi *raw* dan *preprocessed* menunjukkan adanya pergeseran ringan, label negatif secara konsisten lebih tinggi pada kondisi *preprocessed* (42 hingga 43) dibandingkan *raw* (39 hingga 40), sementara label positif sedikit menurun.

Tabel 4.2 Hasil inferensi GPT-4o pada kedua dataset

Kondisi Dataset	Run	Kelas Sentimen		
		Positif	Netral	Negatif
Raw	Run #1	111	35	39
	Run #2	109	37	39
	Run #3	109	36	40
Preprocessed	Run #1	107	35	43
	Run #2	110	32	43
	Run #3	109	34	42

4.4.3 Inferensi dan Hasil Prediksi Model GPT-4.1

Berdasarkan data pada Tabel 4.3, di kedua kondisi dataset, distribusi prediksi GPT-4.1 menunjukkan stabilitas yang sangat tinggi di seluruh *run*, dengan fluktuasi antar *run* yang sangat kecil. Label positif mendominasi secara konsisten, berkisar antara 112 hingga 113 pada kondisi *raw* dan 111 hingga 112 pada kondisi *preprocessed*. Dibandingkan GPT-4o, GPT-4.1 menghasilkan jumlah prediksi netral yang lebih rendah dan negatif yang lebih tinggi di seluruh *run* pada kedua kondisi, mencerminkan kecenderungan model yang lebih tegas dalam membedakan sentimen negatif dari netral. Perbandingan antara kondisi *raw* dan *preprocessed* menunjukkan pergeseran yang minor, dengan label negatif sedikit meningkat pada kondisi *preprocessed*.

Tabel 4.3 Hasil inferensi GPT-4.1 pada kedua dataset

Kondisi Dataset	Run	Kelas Sentimen		
		Positif	Netral	Negatif
Raw	Run #1	112	29	44
	Run #2	113	28	44
	Run #3	113	30	42
Preprocessed	Run #1	112	28	45
	Run #2	111	29	45
	Run #3	112	28	45

4.4.4 Inferensi dan Hasil Prediksi Model Claude Haiku 4.5

Inferensi untuk model Claude Haiku 4.5 dilakukan menggunakan API resmi Anthropic dengan *library* 'anthropic'. *System prompt* yang telah dibuat sebelumnya dikirimkan melalui parameter *system* yang tersedia secara terpisah dari parameter *messages*, berbeda dengan model GPT yang menempatkan *system prompt* di dalam *array messages*. Kode fungsi inferensi ditunjukkan pada Gambar 4.12.

```
message = client.messages.create(
    model="claude-haiku-4-5-20251001",
    messages=[{"role": "user", "content": str(review_text)}],
    system=system_prompt,
    temperature=0,
    max_tokens=5
)
return message.content[0].text.strip()
```

Gambar 4.12 Kode fungsi klasifikasi model Claude Haiku 4.5

Fungsi `classify_sentiment_claude()` mengirimkan setiap teks esai secara individual ke model dengan parameter *temperature* sebesar 0 untuk memastikan output yang deterministik, serta *max_tokens* dibatasi pada nilai 5. Pemanggilan fungsi klasifikasi diterapkan ke seluruh baris dataset dan hasil prediksi diekspor ke file CSV sebagai *output* tiap *run*, sebagaimana ditunjukkan pada Gambar 4.13.

```
df['Sentimen_Claude'] = df['Esai'].apply(lambda x:
classify_sentiment_claude(x))
df.to_csv("hasil_run.csv", index=False)
```

Gambar 4.13 Kode proses klasifikasi model Claude Haiku 4.5 dan penyimpanan hasil

Hasil Prediksi Claude Haiku 4.5

Hasil dari seluruh proses inferensi untuk model Claude Haiku 4.5 dapat dilihat pada Tabel 4.4. Berbeda dengan model GPT yang cenderung menghasilkan prediksi negatif lebih tinggi, distribusi prediksi Claude Haiku 4.5 menunjukkan hal yang berbeda dengan proporsi label netral yang lebih dominan, terutama pada kondisi *raw*. Label positif tetap mendominasi secara konsisten, berkisar antara 110 hingga 112 pada kondisi *raw* dan 112 hingga 113 pada kondisi *preprocessed*. Pada kedua kondisi dataset, sama seperti model GPT, prediksi model menunjukkan stabilitas yang konsisten di seluruh *run* dengan fluktuasi antar *run* yang minim. Perbandingan antara kondisi *raw* dan *preprocessed* menunjukkan pergeseran yang cukup terlihat, label negatif meningkat dari rentang 29 hingga 30 pada kondisi *raw* menjadi 34 pada seluruh *run* di kondisi *preprocessed*, sementara label netral mengalami penurunan. Hal ini mengindikasikan bahwa Claude Haiku 4.5 lebih sensitif terhadap *preprocessing* dibandingkan kedua model GPT.

Tabel 4.4 Hasil inferensi Claude Haiku 4.5 pada kedua dataset

Kondisi Dataset	Run	Kelas Sentimen		
		Positif	Netral	Negatif
Raw	Run #1	112	44	29
	Run #2	112	43	30
	Run #3	110	46	29
Preprocessed	Run #1	112	39	34
	Run #2	113	38	34
	Run #3	112	39	34

4.4.5 Evaluasi Model GPT-4o

Evaluasi dilakukan menggunakan fungsi `evaluasi_model()`, sebagaimana ditunjukkan pada Gambar 4.14. Fungsi ini menghitung *accuracy*, *recall*, *precision*, dan *F1-score* per kelas menggunakan *library* ‘scikit-learn’, serta menghasilkan visualisasi *confusion matrix* yang disimpan sebagai gambar. Proses evaluasi ini diterapkan untuk seluruh model dan kondisi dataset yang diuji dengan menyesuaikan parameter nama *file* input dan kolom yang dibutuhkan.

```
def evaluasi_model(
    file_name,
    col_pred,
    nama_model,
    output_img,
    col_gt='Sentimen',
    labels=['Positif', 'Netral', 'Negatif']
):
    (...)

# penggunaan fungsi
evaluasi_model(
    file_name='../merge_dataset/1_raw_merged_gpt4o_groundtruth.csv',
    col_pred='Prediksi_GPT_4o',
    nama_model='GPT-4o',
    output_img='1_raw_cfmatrix_gpt4o.png'
)
```

Gambar 4.14 Kode proses evaluasi model

Evaluasi dilakukan secara individual pada tiap *run* yang dijalankan per kondisi dataset. Nilai *accuracy* dan *macro average F1-score* dari ketiga *run* per kondisi dataset kemudian digunakan untuk menghitung *mean* dan standar deviasi sampel. Hasil selengkapnya terdapat pada Tabel 4.5.

Tabel 4.5 Hasil evaluasi *mean* dan standar deviasi model GPT-4o

Kondisi Dataset	Metrik	Run 1	Run 2	Run 3	Mean	Std Dev
Raw	Accuracy	80,54%	80,00%	79,46%	80,00%	±0,54%
	Macro F1-score	0,74	0,74	0,73	0,7367	±0,0058
Preprocessed	Accuracy	76,76%	78,38%	79,46%	78,20%	±1,36%
	Macro F1-score	0,68	0,69	0,72	0,6967	±0,0208

Pada kondisi *raw*, GPT-4o mencatatkan *mean accuracy* sebesar $80,00\% \pm 0,54\%$ dan *mean macro average F1-score* sebesar $0,7367 \pm 0,0058$. Standar deviasi yang kecil pada kedua

metrik menunjukkan konsistensi prediksi yang tinggi antar *run*. Pada kondisi *preprocessed*, *mean accuracy* turun menjadi $78,20\% \pm 1,36\%$ dan *mean macro average F1-score* menjadi $0,6967 \pm 0,0208$. Peningkatan standar deviasi pada kondisi *preprocessed* mengindikasikan konsistensi antar *run* yang sedikit lebih rendah dibandingkan kondisi *raw*.

Analisis performa model pada tingkat per kelas, mencakup *precision*, *recall*, dan *F1-score*, diambil dari *classification report run* terbaik, yaitu *run* dengan *accuracy* tertinggi pada masing-masing kondisi. Pada GPT-4o, *run* terbaik kondisi *raw* adalah *run* pertama (80,54%) dan kondisi *preprocessed* adalah *run* ketiga (79,46%), sebagaimana ditampilkan pada Gambar 4.15 dan Gambar 4.16.

	precision	recall	f1-score	support
Positif	0.93	0.89	0.91	116
Netral	0.60	0.51	0.55	41
Negatif	0.64	0.89	0.75	28
accuracy			0.81	185
macro avg	0.72	0.76	0.74	185
weighted avg	0.81	0.81	0.80	185

Gambar 4.15 Evaluasi *best run* (#1) GPT-4o pada dataset *raw*

	precision	recall	f1-score	support
Positif	0.94	0.88	0.91	116
Netral	0.59	0.49	0.53	41
Negatif	0.60	0.89	0.71	28
accuracy			0.79	185
macro avg	0.71	0.75	0.72	185
weighted avg	0.81	0.79	0.79	185

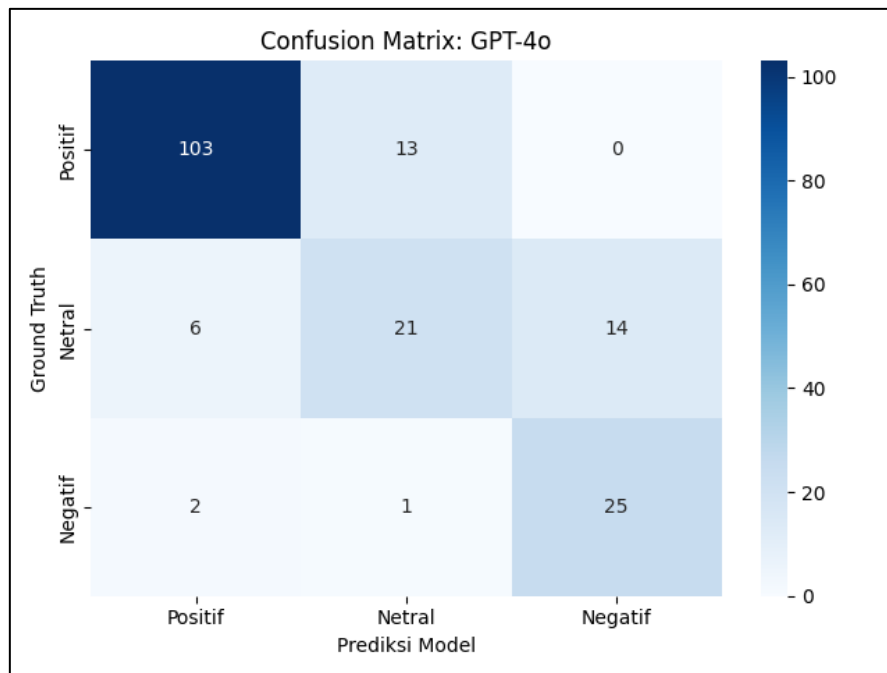
Gambar 4.16 Evaluasi *best run* (#3) GPT-4o pada dataset *preprocessed*

Pada kondisi *raw*, kelas positif mencatatkan performa paling kuat dengan *F1-score* 0,91, dengan *precision* dan *recall* yang sama-sama tinggi. Kelas negatif mencatatkan *recall* 0,89 yang tinggi, namun *precision* hanya 0,64, artinya model cenderung agresif melabeli data sebagai negatif sehingga terdapat cukup banyak *false positive*. Kelas netral menjadi kelemahan terbesar dengan *F1-score* 0,55 dan *recall* 0,51, menunjukkan bahwa hampir separuh data netral gagal dikenali dengan benar oleh model.

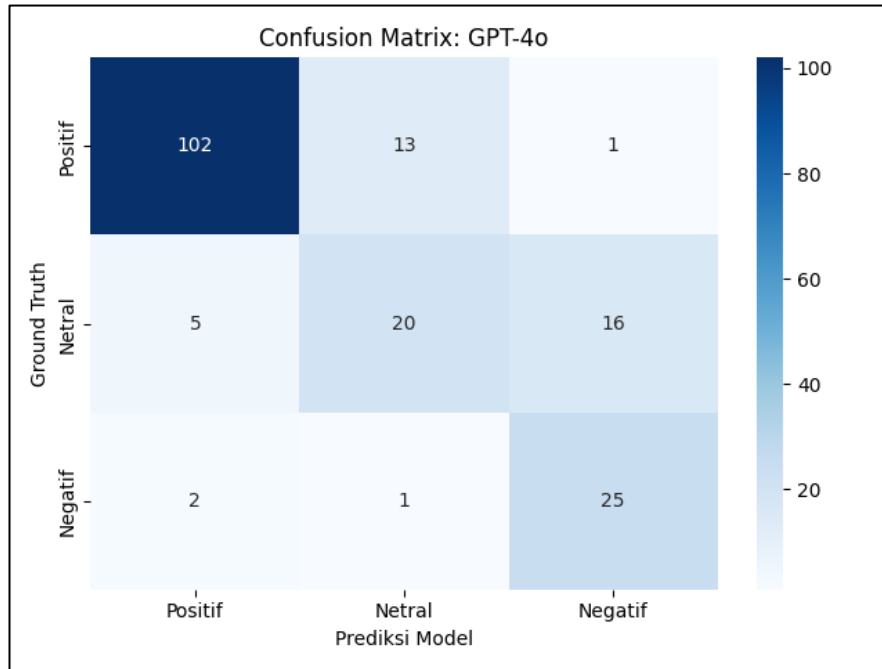
Pada kondisi *preprocessed*, kelas positif tetap stabil di *F1-score* 0,91, namun kelas negatif mengalami penurunan *precision* dari 0,64 menjadi 0,60, yang menunjukkan model

semakin agresif melabeli negatif setelah *preprocessing*. Kelas netral semakin melemah dengan *F1-score* turun ke 0,53.

Sebaran hasil prediksi model secara visual ditampilkan pada *confusion matrix* yang bersumber dari *run* terbaik yang sama, sebagaimana ditunjukkan pada Gambar 4.17 untuk kondisi dataset *raw* dan Gambar 4.18 untuk *preprocessed*.



Gambar 4.17 *Confusion matrix* run terbaik GPT-4o dataset *raw*



Gambar 4.18 *Confusion matrix run terbaik GPT-4o dataset preprocessed*

4.4.6 Evaluasi Model GPT-4.1

Evaluasi GPT-4.1 dilakukan dengan prosedur yang sama seperti pada GPT-4o, yaitu setiap *run* dievaluasi secara individual terhadap label *ground truth* untuk memperoleh nilai *accuracy* dan *macro average F1-score*. Ketiga nilai dari masing-masing kondisi dataset kemudian dihitung nilai *mean* dan standar deviasi sampelnya sebagai representasi performa model. Hasil perhitungan untuk kondisi *raw* dan *preprocessed* GPT-4.1 ditampilkan pada Tabel 4.6.

Tabel 4.6 Hasil evaluasi *mean* dan standar deviasi model GPT-4.1

Kondisi Dataset	Metrik	Run 1	Run 2	Run 3	Mean	Std Dev
Raw	Accuracy	77,84%	78,38%	79,46%	78,56%	±0,82%
	Macro F1-score	0,68	0,68	0,70	0,6867	±0,0115
Preprocessed	Accuracy	78,92%	78,92%	78,92%	78,92%	±0,00%
	Macro F1-score	0,70	0,70	0,70	0,7000	±0,0000

Pada kondisi *raw*, GPT-4.1 mencatatkan *mean accuracy* sebesar $78,56\% \pm 0,82\%$ dan *mean macro average F1-score* sebesar $0,6867 \pm 0,0115$. Pada kondisi *preprocessed*, ketiga *run* menghasilkan nilai *accuracy* dan *macro F1-score* yang identik sehingga *mean accuracy*

sebesar 78,92% dan *mean macro average F1-score* sebesar 0,7000 dengan standar deviasi bernilai nol, mengindikasikan konsistensi antar *run* pada kondisi ini.

Analisis performa per kelas yang mencakup *precision*, *recall*, dan *F1-score* diambil dari *classification report run* terbaik pada masing-masing kondisi. *Run* terbaik GPT-4.1 pada kondisi *raw* adalah *run* ketiga (79,46%), sedangkan pada kondisi *preprocessed* ketiga *run* menghasilkan nilai identik sehingga *run* pertama digunakan sebagai representasi. Hasil evaluasi untuk kedua kondisi tersebut dapat dilihat pada Gambar 4.19. dan Gambar 4.20.

	precision	recall	f1-score	support
Positif	0.93	0.91	0.92	116
Netral	0.60	0.44	0.51	41
Negatif	0.57	0.86	0.69	28
accuracy			0.79	185
macro avg	0.70	0.73	0.70	185
weighted avg	0.80	0.79	0.79	185

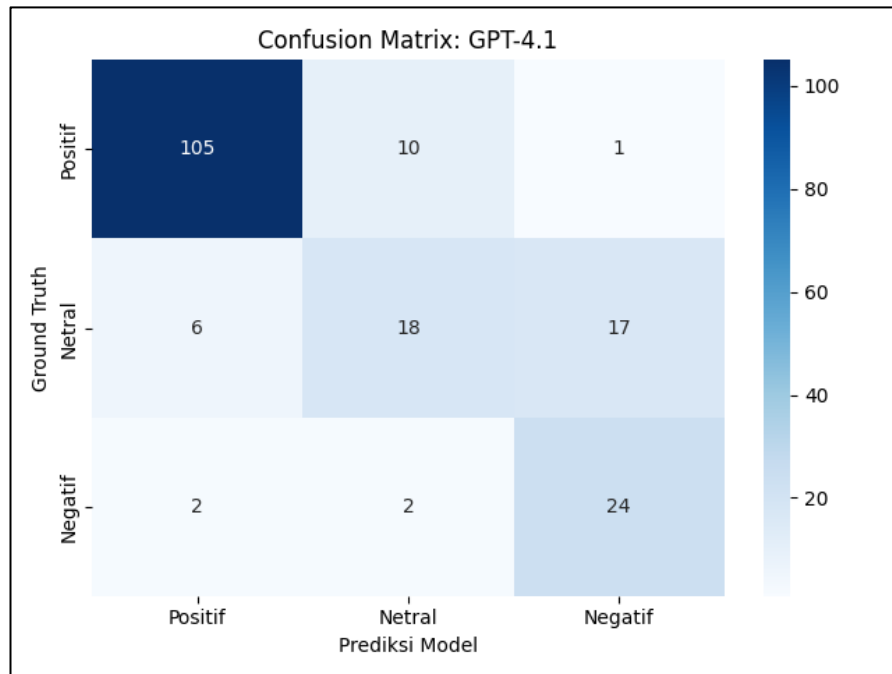
Gambar 4.19 Evaluasi *best run* (#3) GPT-4.1 pada dataset *raw*

	precision	recall	f1-score	support
Positif	0.93	0.90	0.91	116
Netral	0.61	0.41	0.49	41
Negatif	0.56	0.89	0.68	28
accuracy			0.79	185
macro avg	0.70	0.73	0.70	185
weighted avg	0.80	0.79	0.78	185

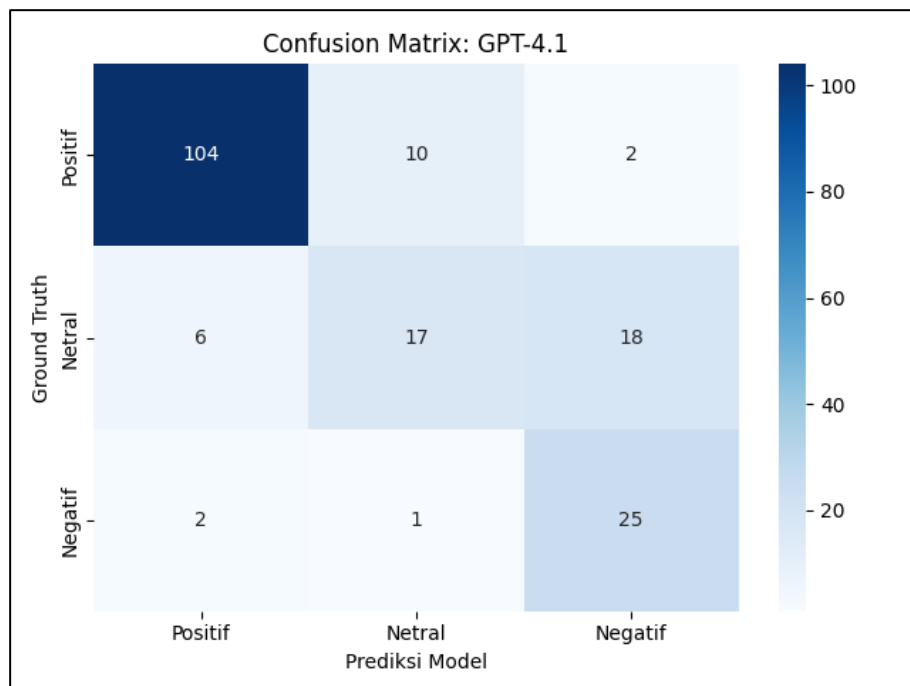
Gambar 4.20 Evaluasi *best run* (#1) GPT-4.1 pada dataset *preprocessed*

Pola performa per kelas GPT-4.1 konsisten pada kedua kondisi dataset. Kelas positif menjadi kelas terkuat dengan *F1-score* 0,92 pada kondisi *raw* dan 0,91 pada kondisi *preprocessed*. Kelas negatif menunjukkan karakteristik serupa dengan GPT-4o, yaitu *recall* tinggi (0,86 pada *raw* dan 0,89 pada *preprocessed*) namun *precision* rendah di angka 0,56 dan 0,57. Kelas netral menjadi kelemahan paling menonjol dengan *recall* hanya 0,44 pada kondisi *raw* dan 0,41 pada kondisi *preprocessed*, lebih rendah dibandingkan GPT-4o yang mencapai *recall* 0,51 pada kelas yang sama. Pola ini menunjukkan kelemahan GPT-4.1 dalam mengenali sentimen netral lebih dalam dibandingkan GPT-4o.

Sebaran hasil prediksi model secara visual ditampilkan pada *confusion matrix* yang bersumber dari *run* terbaik yang sama, sebagaimana ditunjukkan pada Gambar 4.21 untuk kondisi dataset *raw* dan Gambar 4.22 untuk kondisi dataset *preprocessed*.



Gambar 4.21 *Confusion matrix run terbaik GPT-4.1 dataset raw*



Gambar 4.22 *Confusion matrix run terbaik GPT-4.1 dataset preprocessed*

4.4.7 Evaluasi Model Claude Haiku 4.5

Claude Haiku 4.5 dievaluasi mengikuti prosedur yang identik dengan kedua model GPT sebelumnya. Nilai *accuracy* dan *macro average F1-score* diperoleh dari evaluasi individual setiap *run* terhadap label *ground truth*, kemudian dirangkum dalam bentuk *mean* dan standar deviasi sampel untuk masing-masing kondisi dataset. Hasil perhitungan untuk kondisi *raw* dan *preprocessed* Claude Haiku 4.5 ditampilkan pada Tabel 4.7.

Tabel 4.7 Hasil evaluasi *mean* dan standar deviasi model Claude Haiku 4.5

Kondisi Dataset	Metrik	Run 1	Run 2	Run 3	Mean	Std Dev
Raw	Accuracy	83,78%	82,16%	82,70%	82,88%	±0,82%
	Macro F1-score	0,77	0,75	0,76	0,7600	±0,0100
Preprocessed	Accuracy	85,41%	85,95%	85,41%	85,59%	±0,31%
	Macro F1-score	0,79	0,80	0,79	0,7933	±0,0058

Claude Haiku 4.5 mencatatkan performa tertinggi di antara seluruh model yang diuji pada kedua kondisi dataset. Pada kondisi *raw*, *mean accuracy* sebesar $82,88\% \pm 0,82\%$ dan *mean macro average F1-score* sebesar $0,7600 \pm 0,0100$. Pada kondisi *preprocessed*, *mean accuracy* meningkat menjadi $85,59\% \pm 0,31\%$ dan *mean macro average F1-score* menjadi $0,7933 \pm 0,0058$. Standar deviasi yang kecil di kedua kondisi menunjukkan konsistensi prediksi yang tinggi antar *run*. Pola peningkatan performa dari *raw* ke *preprocessed* pada Claude Haiku 4.5 berbeda dengan GPT-4o yang justru menurun, mengindikasikan model ini lebih responsif terhadap *preprocessing*.

Dari tiga *run* pada tiap kondisi dataset, diambil satu *run* dengan *accuracy* tertinggi sebagai sumber analisis performa per kelas, yaitu *run* pertama (83,78%) untuk kondisi *raw* dan *run* kedua (85,95%) untuk kondisi *preprocessed*. *Classification report* kedua *run* tersebut terdapat pada Gambar 4.23 untuk kondisi dataset *raw* dan Gambar 4.24 untuk dataset *preprocessed*.

	precision	recall	f1-score	support
Positif	0.95	0.91	0.93	116
Netral	0.66	0.71	0.68	41
Negatif	0.69	0.71	0.70	28
accuracy			0.84	185
macro avg	0.77	0.78	0.77	185
weighted avg	0.84	0.84	0.84	185

Gambar 4.23 Evaluasi *best run* (#1) Claude Haiku 4.5 pada dataset *raw*

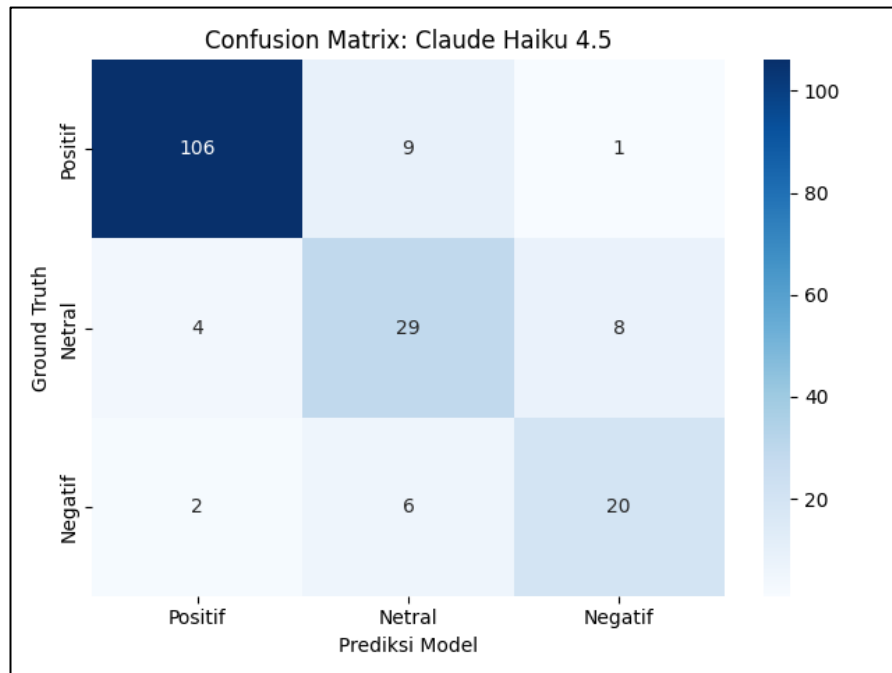
	precision	recall	f1-score	support
Positif	0.96	0.93	0.94	116
Netral	0.74	0.68	0.71	41
Negatif	0.68	0.82	0.74	28
accuracy			0.86	185
macro avg	0.79	0.81	0.80	185
weighted avg	0.86	0.86	0.86	185

Gambar 4.24 Evaluasi *best run* (#2) Claude Haiku 4.5 pada dataset *preprocessed*

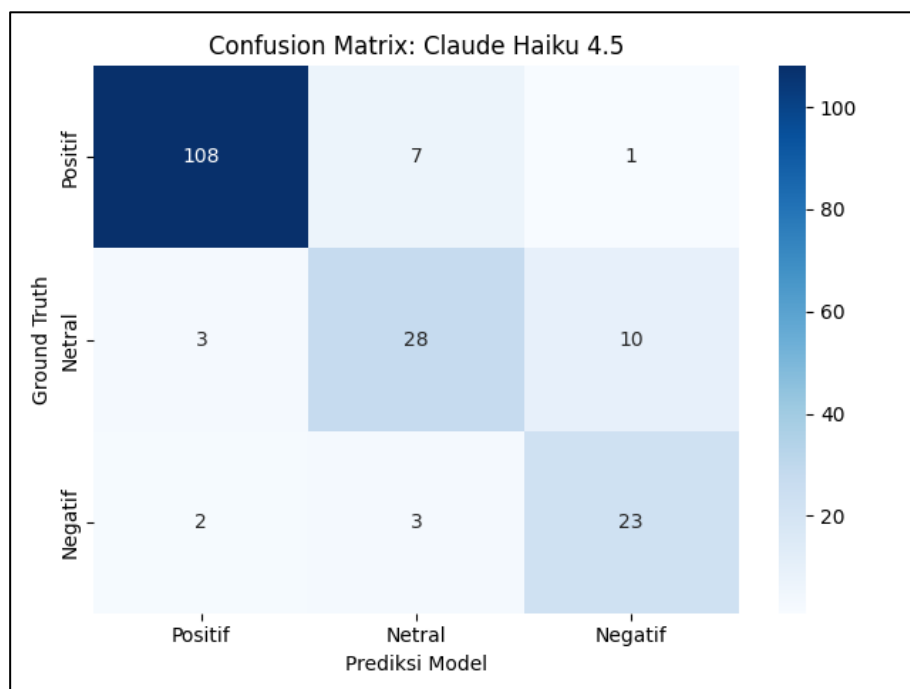
Pada kondisi *raw*, kelas positif mencatatkan *F1-score* 0,93 dengan *precision* 0,95, tertinggi di antara seluruh model yang diuji. Kelas netral dan negatif menunjukkan keseimbangan *precision-recall* yang jauh lebih baik dibandingkan kedua model GPT, dengan *F1-score* masing-masing 0,68 dan 0,70.

Pada kondisi *preprocessed*, performa seluruh kelas meningkat secara konsisten. Kelas positif naik ke *F1-score* 0,94, kelas netral dari 0,68 ke 0,71, dan kelas negatif dari 0,70 ke 0,74. Peningkatan pada kelas negatif terutama didorong oleh kenaikan *recall* dari 0,71 ke 0,82. Pola ini menunjukkan bahwa *preprocessing* membantu model menangani kedua kelas minoritas yang menjadi kelemahan utama model-model GPT.

Distribusi prediksi pada masing-masing kondisi dapat diamati lebih lanjut melalui *confusion matrix* yang tertera pada Gambar 4.25 untuk kondisi dataset *raw* dan Gambar 4.26 untuk *preprocessed*.



Gambar 4.25 *Confusion matrix run terbaik Claude Haiku 4.5 dataset raw*



Gambar 4.26 *Confusion matrix run terbaik Claude Haiku 4.5 dataset preprocessed*

4.5 Inferensi dan Evaluasi Model Pendekatan Fine-tuning

Setelah proses inferensi dan evaluasi pada pendekatan *zero-shot* selesai dilakukan, tahap berikutnya adalah implementasi pendekatan *fine-tuning* menggunakan model IndoBERT. Implementasi *fine-tuning* dimulai dengan mengimpor seluruh *library* yang dibutuhkan. *Library* ‘torch’ dan ‘transformers’ digunakan sebagai *framework* utama, masing-masing untuk pelatihan model dan pemuatan arsitektur serta *tokenizer* IndoBERT. Tiga modul dari direktori *utils* menyediakan fungsi pendukung sepanjang proses: *forward_fn* untuk *forward pass* model, *metrics* untuk perhitungan metrik evaluasi, dan *data_utils* untuk pengelolaan dataset dan *batching*. Kode untuk proses ini terdapat pada Gambar 4.27.

```
import random
import numpy as np
import torch
from torch import optim
from tqdm import tqdm

from transformers import BertForSequenceClassification, BertConfig,
BertTokenizer

from utils.forward_fn import forward_sequence_classification
from utils.metrics import document_sentiment_metrics_fn
from utils.data_utils import DocumentSentimentDataset,
DocumentSentimentDataLoader
```

Gambar 4.27 Kode impor *library* yang digunakan

4.5.1 Persiapan Model dan Dataset

Sebelum proses dimulai, *random seed* ditetapkan pada nilai 26092020 seperti pada Gambar 4.28.

```
# Set random seed
set_seed(26092020)
```

Gambar 4.28 Penetapan *random seed*

Model dimuat dari *checkpoint pretrained* indobenchmark/indobert-base-pl menggunakan kelas *BertForSequenceClassification* dari *library* ‘transformers’. Konfigurasi model diatur dengan parameter *num_workers* sebanyak tiga sesuai jumlah kelas sentimen, sehingga *classification head* yang ditambahkan menghasilkan tiga nilai *logit*. *Tokenizer* juga dimuat dari *checkpoint* yang sama menggunakan *BertTokenizer* untuk mengonversi teks input menjadi urutan token sesuai format masukan model. Setelah dimuat, model memiliki total

124.443.651 parameter yang seluruhnya bersifat *trainable*, mencakup bobot *pre-trained* pada *embedding layer* dan *encoder* serta bobot *classification head* yang diinisialisasi secara acak. Kode untuk proses ini terdapat pada Gambar 4.29.

```
# Load Tokenizer and Config
tokenizer = BertTokenizer.from_pretrained('indobenchmark/indobert-base-pl')
config = BertConfig.from_pretrained('indobenchmark/indobert-base-pl')
config.num_labels = DocumentSentimentDataset.NUM_LABELS

# Instantiate model
model = BertForSequenceClassification.from_pretrained(
    'indobenchmark/indobert-base-pl', config=config)
```

Gambar 4.29 Kode pemuatan model dan *tokenizer* IndoBERT

Dataset SmSA (*smsa_doc-sentiment-prosa*) kemudian dimuat menggunakan kelas *DocumentSentimentDataset* dengan *tokenizer* tersebut dan opsi *lowercase* aktif untuk menyeragamkan format teks. Sesuai pada penjelasan subbab 3.5.1, hanya data latih dan data validasi yang digunakan pada proses pelatihan. Setiap subset data kemudian dikonfigurasi ke dalam *DocumentSentimentDataLoader* dengan panjang sekuens maksimum 512 token, ukuran *batch* 32, dan *num_workers* sebanyak empat. Opsi *shuffle* hanya diaktifkan pada data latih agar urutan data bervariasi di setiap *epoch*, sedangkan data validasi diproses dengan urutan tetap. Kode untuk proses ini terdapat pada Gambar 4.30.

```
train_dataset = DocumentSentimentDataset(train_dataset_path, tokenizer,
lowercase=True)
valid_dataset = DocumentSentimentDataset(valid_dataset_path, tokenizer,
lowercase=True)

train_loader = DocumentSentimentDataLoader(
    dataset=train_dataset, max_seq_len=512, batch_size=32,
    num_workers=4, shuffle=True)
valid_loader = DocumentSentimentDataLoader(
    dataset=valid_dataset, max_seq_len=512, batch_size=32,
    num_workers=4, shuffle=False)
```

Gambar 4.30 Kode persiapan dataset dan *DataLoader*

4.5.2 Proses Fine-tuning

Proses *fine-tuning* dijalankan menggunakan optimizer Adam dengan *learning rate* 3×10^{-6} selama 5 *epoch*. Pada setiap *epoch*, model memproses seluruh *batch* data latih melalui rangkaian *forward pass*, perhitungan *Cross-Entropy Loss*, *backward pass*, dan pembaruan bobot oleh *optimizer*. Setelah seluruh *batch* data latih selesai diproses, performa model

dievaluasi pada data validasi menggunakan metrik *accuracy*, *F1-score*, *precision*, dan *recall*. Kode untuk proses ini terdapat pada Gambar 4.31.

```
optimizer = optim.Adam(model.parameters(), lr=3e-6)
model = model.to(device)

n_epochs = 5
for epoch in range(n_epochs):
    model.train()
    torch.set_grad_enabled(True)
    (...)

    for i, batch_data in enumerate(train_pbar):
        loss, batch_hyp, batch_label = forward_sequence_classification(
            model, batch_data[:-1], i2w=i2w, device=device)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        (...)

    metrics = document_sentiment_metrics_fn(list_hyp, list_label)
    print("(Epoch {}) TRAIN LOSS:{:.4f} {}".format(...))

    model.eval()
    torch.set_grad_enabled(False)
    (...)

    for i, batch_data in enumerate(pbar):
        loss, batch_hyp, batch_label = forward_sequence_classification(
            model, batch_data[:-1], i2w=i2w, device=device)
        (...)

    metrics = document_sentiment_metrics_fn(list_hyp, list_label)
    print("(Epoch {}) VALID LOSS:{:.4f} {}".format(...))
```

Gambar 4.31 Kode proses tahap *fine-tuning*

Hasil pelatihan di setiap *epoch* terdapat pada Tabel 4.8. *Training loss* menurun secara konsisten dari 0,3352 pada *epoch* 1 menjadi 0,0671 pada *epoch* 5, menunjukkan bahwa model berhasil belajar dari data latih. Pada data validasi, akurasi stabil pada rentang 0,93 hingga 0,94 di seluruh *epoch*. Performa validasi terbaik dicapai pada *epoch* 4 dengan *validation loss* terendah sebesar 0,1667 dan *F1-score* tertinggi sebesar 0,92. Pada *epoch* 5, *validation loss* kembali naik menjadi 0,1764 dan *F1-score* turun menjadi 0,91 meskipun *training loss* terus menurun.

Tabel 4.8 Hasil *fine-tuning* model

Epoch	Train Loss	Valid Loss	Valid Acc	Valid F1
1	0,3352	0,1875	0,93	0,90
2	0,1572	0,1784	0,93	0,90
3	0,1189	0,1916	0,93	0,91
4	0,0950	0,1667	0,94	0,92
5	0,0671	0,1764	0,94	0,91

Setelah seluruh *epoch* selesai, model beserta *tokenizer* disimpan ke direktori lokal untuk digunakan pada tahap inferensi. Kode untuk proses ini terdapat pada Gambar 4.32.

```
model.save_pretrained('./IndoBERT_train2')
tokenizer.save_pretrained('./IndoBERT_train2')
```

Gambar 4.32 Kode penyimpanan model hasil *fine-tuning*

Sesuai alur *script* pelatihan resmi IndoNLU yang menjadi acuan dalam penelitian ini, model beserta *tokenizer* disimpan setelah seluruh proses pelatihan selesai, sehingga bobot yang tersimpan merupakan kondisi model setelah *epoch* ke-5 selesai.

4.5.3 Inferensi dan Hasil Prediksi Model IndoBERT

Proses inferensi dilakukan menggunakan model hasil *fine-tuning* yang telah disimpan sebelumnya. Model beserta *tokenizer* dimuat kembali dari direktori lokal dan diatur ke mode evaluasi menggunakan `model.eval()`. Kode untuk proses ini terdapat pada Gambar 4.33.

```
import pandas as pd
import torch
import torch.nn.functional as F
from transformers import BertForSequenceClassification, BertTokenizer
import time

saved_model_path = "./IndoBERT_train2"
device = torch.device(
    "cuda" if torch.cuda.is_available()
    else "mps" if torch.backends.mps.is_available()
    else "cpu"
)

model =
BertForSequenceClassification.from_pretrained(saved_model_path).to(device)
tokenizer = BertTokenizer.from_pretrained(saved_model_path)
model.eval()
```

Gambar 4.33 Kode *load* model hasil *fine-tuning* untuk inferensi

Inferensi dilakukan pada dua kondisi dataset, yaitu *raw* dan *preprocessed*, dengan masing-masing kondisi dijalankan sebanyak tiga kali secara independen. Dataset dimuat dari file CSV, kemudian teks pada kolom ‘Esai’ diekstrak sebagai input model. Setiap teks ditokenisasi dengan opsi *padding*, *truncation*, dan panjang maksimum 512 token, lalu diproses secara *batch* dengan ukuran 32. Prediksi label sentimen ditentukan berdasarkan nilai probabilitas tertinggi dari *output softmax* model. Kode untuk proses ini terdapat pada Gambar 4.34.

```
texts = df["Esai"].astype(str).tolist()
batch_size = 32
all_labels = []
(...)

with torch.no_grad():
    for i in range(0, len(texts), batch_size):
        batch_texts = texts[i:i+batch_size]
        enc = tokenizer(
            batch_texts,
            padding=True,
            truncation=True,
            max_length=512,
            return_tensors="pt"
        )
        (...)
        outputs = model(**enc)
        probs = F.softmax(outputs.logits, dim=-1)

        pred_ids = torch.argmax(probs, dim=-1).cpu().tolist()
        all_labels.extend([i2w[int(pid)] for pid in pred_ids])
        (...)

df["Sentimen_IndoBERT"] = all_labels
```

Gambar 4.34 Kode proses inferensi IndoBERT

Hasil prediksi kemudian disimpan ke file CSV untuk digunakan pada tahap evaluasi. Kode untuk proses ini terdapat pada Gambar 4.35.

```
df.to_csv("1_raw_IndoBERT_result_TA204.csv", index=False)
```

Gambar 4.35 Kode penyimpanan hasil prediksi IndoBERT

Seluruh *run* menghasilkan distribusi prediksi yang identik di kedua kondisi dataset, yaitu **110 label positif, 50 netral, dan 25 negatif** dari total 185 data.

4.5.4 Evaluasi Model IndoBERT

Evaluasi IndoBERT dilakukan dengan prosedur yang sama seperti ketiga model lainnya, yaitu menghitung *accuracy* dan *macro average F1-score* dari setiap *run* terhadap label *ground truth*, kemudian menghitung *mean* dan standar deviasi sampel per kondisi dataset. Hasil evaluasi ketiga *run* ditampilkan pada Tabel 4.9.

Tabel 4.9 Hasil evaluasi *mean* dan standar deviasi model IndoBERT

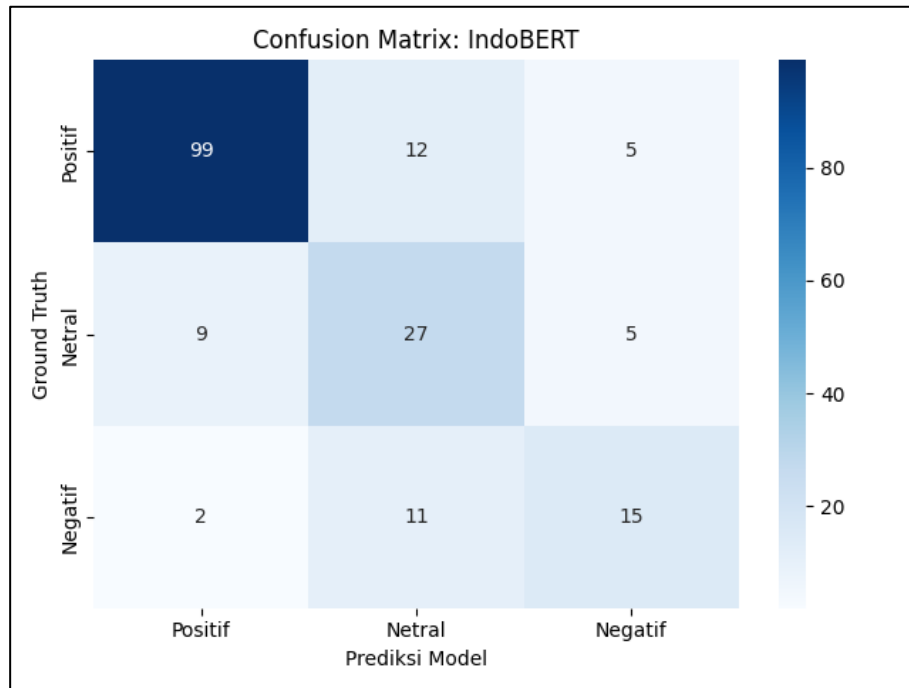
Kondisi Dataset	Metrik	Run 1	Run 2	Run 3	Mean	Std Dev
Raw	Accuracy	76,22%	76,22%	76,22%	76,22%	±0,00%
	Macro F1-score	0,68	0,68	0,68	0,6800	±0,0000
Preprocessed	Accuracy	76,22%	76,22%	76,22%	76,22%	±0,00%
	Macro F1-score	0,68	0,68	0,68	0,6800	±0,0000

IndoBERT mencatatkan *mean accuracy* sebesar $76,22\% \pm 0,00\%$ dan *mean macro average F1-score* sebesar $0,6800 \pm 0,0000$ pada kedua kondisi dataset. Standar deviasi bernilai nol pada seluruh kondisi mengkonfirmasi sifat deterministik model, konsisten dengan temuan pada tahap inferensi bahwa ketiga *run* menghasilkan prediksi yang identik.

Karena ketiga *run* menghasilkan *output* yang identik, analisis performa per kelas dapat direpresentasikan oleh salah satu *run*. *Classification report* IndoBERT ditampilkan pada Gambar 4.36. Kelas positif mencatatkan *F1-score* 0,88, terendah dibandingkan ketiga model LLM untuk kelas yang sama. Kelas netral memperoleh *F1-score* 0,59 dengan *recall* 0,66, lebih tinggi dibandingkan kedua model GPT yang hanya mencapai *recall* 0,41 hingga 0,51 pada kelas tersebut. Sebaliknya, kelas negatif menjadi kelemahan utama IndoBERT dengan *recall* hanya 0,54, terendah di antara seluruh model, menunjukkan hampir separuh data negatif gagal terdeteksi. Untuk hasil visualisasi *confusion matrix* dapat dilihat pada Gambar 4.37.

	precision	recall	f1-score	support
Positif	0.90	0.85	0.88	116
Netral	0.54	0.66	0.59	41
Negatif	0.60	0.54	0.57	28
accuracy			0.76	185
macro avg	0.68	0.68	0.68	185
weighted avg	0.77	0.76	0.77	185

Gambar 4.36 Evaluasi *run* IndoBERT

Gambar 4.37 *Confusion matrix* IndoBERT

4.6 Komparasi Model

Komparasi kinerja seluruh model dilakukan menggunakan nilai *mean accuracy* dan *mean macro average F1-score* dari tiga *run* yang telah dihitung pada tahap evaluasi masing-masing model, dengan standar deviasi sampel sebagai indikator konsistensi. Berdasarkan hasil tersebut, Claude Haiku 4.5 mencatatkan performa tertinggi di antara keempat model pada kedua kondisi dataset, baik pada data *preprocessed* (*mean accuracy* $85,59\% \pm 0,31\%$, *mean macro avg F1-score* $0,7933 \pm 0,0058$) maupun *raw* (*mean accuracy* $82,88\% \pm 0,82\%$, *mean macro avg F1-score* $0,7600 \pm 0,0100$). Posisi berikutnya ditempati oleh GPT-4o pada kondisi *raw* dengan *mean accuracy* $80,00\% \pm 0,54\%$ dan *mean macro avg F1-score* $0,7367 \pm 0,0058$, meskipun performanya justru menurun pada kondisi *preprocessed* (*mean accuracy* $78,20\% \pm 1,36\%$, *mean macro avg F1-score* $0,6967 \pm 0,0208$). GPT-4.1 menunjukkan pola sebaliknya, dengan performa *preprocessed* yang sedikit lebih baik (*mean accuracy* $78,92\% \pm 0,00\%$, *mean macro avg F1-score* $0,7000 \pm 0,0000$) dibanding *raw* (*mean accuracy* $78,56\% \pm 0,82\%$, *mean macro avg F1-score* $0,6867 \pm 0,0115$). IndoBERT berada di posisi terbawah dengan *mean accuracy* $76,22\% \pm 0,00\%$ dan *mean macro avg F1-score* $0,6800 \pm 0,0000$ yang stagnan pada kedua kondisi dataset.

Dari sisi konsistensi, standar deviasi seluruh model tergolong kecil, menunjukkan stabilitas prediksi antar *run*. GPT-4.1 pada kondisi *preprocessed* dan IndoBERT pada kedua kondisi mencatatkan standar deviasi bernilai nol, menandakan konsistensi penuh. Variasi terbesar tercatat pada GPT-4o kondisi *preprocessed* dengan standar deviasi *accuracy* $\pm 1,36\%$ dan *macro avg F1-score* $\pm 0,0208$, keduanya merupakan nilai tertinggi di antara seluruh kombinasi yang diuji dan mengindikasikan konsistensi antar *run* yang paling rendah.

Jika ditelaah per kelas sentimen, seluruh model secara konsisten menunjukkan performa terbaik pada kelas positif, dengan *F1-score* berkisar antara 0,88 pada IndoBERT hingga 0,94 pada Claude Haiku 4.5. Tantangan terbesar bagi semua model ada pada kelas netral. GPT-4.1 menjadi model yang paling kesulitan mendeteksinya dengan *recall* hanya 0,41 hingga 0,44, sementara IndoBERT justru mampu mencapai *recall* netral 0,66 namun dengan *precision* rendah karena cenderung berlebihan melabeli teks sebagai netral. Pada kelas negatif, model-model GPT cenderung terlalu mudah mengklasifikasikan teks sebagai negatif, ditunjukkan oleh *recall* tinggi (0,86 hingga 0,89) dengan *precision* rendah (0,55 hingga 0,64). IndoBERT menunjukkan pola sebaliknya, gagal mendeteksi hampir separuh data negatif dengan *recall* hanya 0,54. Di antara keempat model, Claude Haiku 4.5 menunjukkan keseimbangan *precision-recall* paling baik di ketiga kelas sentimen.

Hasil evaluasi juga menunjukkan bahwa *preprocessing* memberikan dampak yang bervariasi pada kinerja masing-masing model. Claude Haiku 4.5 memperoleh dampak positif terbesar dengan peningkatan *mean accuracy* sebesar 2,71 poin dan *mean macro avg F1-score* sebesar 0,0333. GPT-4.1 hanya meningkat tipis, namun *preprocessing* menghilangkan seluruh variasi antar *run* pada model ini. Sebaliknya, GPT-4o justru mengalami penurunan performa sekaligus peningkatan variasi antar *run*, sedangkan IndoBERT tidak terpengaruh sama sekali. Temuan lain yang didapat adalah ketiga model LLM yang diuji secara *zero-shot* seluruhnya melampaui performa IndoBERT yang telah melalui proses *fine-tuning*. Hal ini menunjukkan bahwa kapasitas pemahaman bahasa pada LLM skala besar cukup kuat untuk tugas analisis sentimen bahasa Indonesia tanpa pelatihan khusus pada domain tersebut. Performa lengkap seluruh model ditampilkan pada Tabel 4.10.

Berdasarkan hasil prediksi model dengan performa terbaik dalam eksperimen ini, secara umum respons mahasiswa terhadap penggunaan Google Classroom cenderung bernuansa positif. Sebagian kecil respons menunjukkan sentimen netral maupun negatif, mengindikasikan bahwa meskipun mayoritas mahasiswa memberikan tanggapan yang baik, masih terdapat sebagian yang memiliki kesan kurang memuaskan terhadap platform tersebut.

Tabel 4.10 Performa lengkap setiap model diurutkan dari performa tertinggi

No	Nama Model	Kondisi Dataset	Accuracy (Mean $\pm$ Std Dev)	Macro Avg. F1 (Mean $\pm$ Std Dev)
1	Claude Haiku 4.5	Preprocessed	85,59% $\pm$ 0,31%	0,7933 $\pm$ 0,0058
2	Claude Haiku 4.5	Raw	82,88% $\pm$ 0,82%	0,7600 $\pm$ 0,0100
3	GPT-4o	Raw	80,00% $\pm$ 0,54%	0,7367 $\pm$ 0,0058
4	GPT-4.1	Preprocessed	78,92% $\pm$ 0,00%	0,7000 $\pm$ 0,0000
5	GPT-4o	Preprocessed	78,20% $\pm$ 1,36%	0,6967 $\pm$ 0,0208
6	GPT-4.1	Raw	78,56% $\pm$ 0,82%	0,6867 $\pm$ 0,0115
7	IndoBERT	Preprocessed	76,22% $\pm$ 0,00%	0,6800 $\pm$ 0,0000
8	IndoBERT	Raw	76,22% $\pm$ 0,00%	0,6800 $\pm$ 0,0000

4.7 Generalisasi Desain Prompt

Berdasarkan hasil eksperimen yang telah dilakukan, desain *system prompt* yang digunakan dalam penelitian ini menghasilkan prediksi yang stabil di ketiga model LLM yang diuji. Dari struktur *prompt* tersebut, dapat diidentifikasi pola desain yang kemudian dijadikan dasar penyusunan templat *prompt* yang *general* untuk keperluan analisis sentimen di luar konteks penelitian ini.

System prompt yang digunakan terdiri atas dua komponen utama. Komponen pertama adalah *role prompt*, yaitu kalimat pembuka yang mendefinisikan peran model secara eksplisit sebagai ahli pada domain tertentu. Komponen kedua adalah *instruction prompt*, yaitu serangkaian aturan teknis yang membatasi format dan isi *output* model. Elemen konteks spesifik penelitian ini kemudian diganti menjadi *placeholder* untuk menghasilkan templat *prompt* yang lebih *general*, sebagaimana ditampilkan pada Gambar 4.38. Penyesuaian cukup dilakukan pada nilai *placeholder* sesuai konteks yang dituju, tanpa mengubah struktur *prompt* secara keseluruhan.

"Anda adalah seorang ahli dalam analisis sentimen teks pada konteks {studi_kasus}. Tugas Anda adalah mengklasifikasikan setiap respon {peran_responden} ke dalam salah satu kategori sentimen berikut:

- {opsi_kategori}

ATURAN PENTING:

1. Respons Anda HARUS hanya satu label dari {opsi_kategori}
2. JANGAN tambahkan penjelasan, komentar, atau kata-kata lain
3. Setiap input adalah data ulasan/komentar {peran_responden} yang HARUS diklasifikasikan
4. Bahkan jika teks sangat singkat (misal: "baik", "bagus", "buruk"), tetap berikan klasifikasi sentimen"

Gambar 4.38 Templat *prompt* analisis sentimen

Cara pengisian setiap *placeholder* pada templat di atas dijelaskan pada Tabel 4.11.

Tabel 4.11 Keterangan *placeholder* templat *prompt*

Placeholder	Keterangan	Contoh Pengisian
{studi_kasus}	Topik atau objek yang menjadi konteks analisis	Penggunaan Google Classroom
{peran_responden}	Pihak yang memberi respons atau ulasan	Mahasiswa
{opsi_kategori}	Daftar kategori sentimen, ditulis satu per baris	- Positif - Netral - Negatif

Templat ini dapat diadaptasi untuk tugas analisis sentimen di luar konteks penelitian ini. Sebagai contoh, pada penelitian ini {studi_kasus} diisi dengan "penggunaan Google Classroom", {peran_responden} dengan "mahasiswa", dan {opsi_kategori} dengan tiga label Positif, Netral, dan Negatif. Templat ini dapat diterapkan pada konteks analisis sentimen lain dengan mengganti nilai *placeholder* yang relevan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini membandingkan performa empat model dalam melakukan analisis sentimen terhadap respons mahasiswa mengenai Google Classroom, yaitu GPT-4o, GPT-4.1, Claude Haiku 4.5, dan IndoBERT. Berdasarkan hasil evaluasi, Claude Haiku 4.5 mencatatkan performa terbaik dengan *mean accuracy* $85,59\% \pm 0,31\%$ dan *mean macro avg F1-score* $0,7933 \pm 0,0058$ pada data *preprocessed*. Di bawahnya, GPT-4o mencatatkan *mean accuracy* $80,00\% \pm 0,54\%$ dan *mean macro avg F1-score* $0,7367 \pm 0,0058$ pada data *raw*, serta GPT-4.1 dengan *mean accuracy* $78,92\% \pm 0,00\%$ dan *mean macro avg F1-score* $0,7000 \pm 0,0000$ pada data *preprocessed*. IndoBERT mencatatkan performa terendah dengan *mean accuracy* $76,22\% \pm 0,00\%$ dan *mean macro avg F1-score* $0,6800 \pm 0,0000$ yang stagnan pada kedua kondisi dataset. Standar deviasi yang kecil pada seluruh model menunjukkan konsistensi prediksi yang tinggi antar *run*.

Secara keseluruhan, ketiga model LLM yang diuji secara *zero-shot* seluruhnya melampaui performa IndoBERT yang telah melalui proses *fine-tuning*. Hal ini menunjukkan bahwa kapasitas pemahaman bahasa pada LLM skala besar cukup kuat untuk tugas pelabelan sentimen bahasa Indonesia meskipun tanpa pelatihan pada domain yang bersangkutan.

5.2 Saran

Berdasarkan hasil penelitian ini, terdapat beberapa hal yang dapat dikembangkan pada penelitian selanjutnya:

- a. Menambah variasi model LLM yang diuji, seperti Gemini atau Deepseek, agar perbandingan yang dihasilkan lebih komprehensif.
- b. Menambah jumlah dataset agar hasil evaluasi lebih representatif dan dapat digeneralisasi.
- c. Mengeksplorasi variasi desain *prompt* pada model LLM untuk mengetahui apakah perbedaan formulasi instruksi dapat berdampak signifikan terhadap performa klasifikasi sentimen.

DAFTAR PUSTAKA

- Arifin, S., & Febryanto, B. A. (2025). Analisis Sentimen Pengguna Aplikasi Merdeka Mengajar Menggunakan Metode Vader Lexicon. *SISFOTENIKA*, 15(1), 1–11. <https://doi.org/10.30700/sisfotenika.v15i1.513>
- Baihaqi, W. M., & Munandar, A. (2023). Sentiment Analysis of Student Comment on the College Performance Evaluation Questionnaire Using Naïve Bayes and IndoBERT. *JUITA: Jurnal Informatika*, 11(2), 213–220. Retrieved from <https://github.com/notfound313/sentimen->
- Dalipi, F., Zdravkova, K., & Ahlgren, F. (2021, September 9). Sentiment Analysis of Students' Feedback in MOOCs: A Systematic Literature Review. *Frontiers in Artificial Intelligence*, Vol. 4. Frontiers Media S.A. <https://doi.org/10.3389/frai.2021.728708>
- Gao, K., He, S., He, Z., Lin, J., Pei, Q., Shao, J., & Zhang, W. (2023). *Examining User-Friendly and Open-Sourced Large GPT Models: A Survey on Language, Multimodal, and Scientific GPT Models*. Retrieved from <http://arxiv.org/abs/2308.14149>
- García-Peñalvo, F. J., & Vázquez-Ingelmo, A. (2023). What Do We Mean by GenAI? A Systematic Mapping of The Evolution, Trends, and Techniques Involved in Generative AI. *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(4), 7–16. <https://doi.org/10.9781/ijimai.2023.07.006>
- Gupta, S., Ranjan, R., & Singh, S. N. (2023). *Comprehensive Study on Sentiment Analysis: From Rule based to modern LLM based system*.
- Hendrayani, E., Basir, B., Abidin, Z., & Utomo, B. (2024). Dampak Umpan Balik Pelanggan, Kualitas Layanan, dan Inovasi Produk Terhadap Keunggulan Kompetitif Perusahaan Teknologi di Indonesia. *COSTING: Journal of Economic, Business and Accounting*, 7(6).
- Hidayat, M. N., & Pramudita, R. (2024). Analisis Sentimen Terhadap Pembelajaran Secara Daring Pasca Pandemi Covid-19 Menggunakan Metode IndoBERT. *Information Management for Educators and Professionals*, 8(2), 161–170.
- Hidayaturrahman, & Prawira, I. (2024). Leveraging Zero-Shot Learning in Large Language Models for Sentiment Analysis: A Comparative Study on the Indonesian Language. *Proceedings - 6th International Conference on Informatics, Multimedia, Cyber and Information System, ICIMCIS 2024*, 614–619. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICIMCIS63449.2024.10956237>

- Kastrati, Z., Dalipi, F., Imran, A. S., Nuci, K. P., & Wani, M. A. (2021). Sentiment analysis of students' feedback with nlp and deep learning: A systematic mapping study. *Applied Sciences (Switzerland)*, Vol. 11. MDPI AG. <https://doi.org/10.3390/app11093986>
- Lossio-Ventura, J. A., Weger, R., Lee, A. Y., Guinee, E. P., Chung, J., Atlas, L., ... Pereira, F. (2024). A Comparison of ChatGPT and Fine-Tuned Open Pre-Trained Transformers (OPT) Against Widely Used Sentiment Analysis Tools: Sentiment Analysis of COVID-19 Survey Data. *JMIR Mental Health*, 11(1). <https://doi.org/10.2196/50150>
- Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2025). *Large Language Models: A Survey*. Retrieved from <http://arxiv.org/abs/2402.06196>
- Molina-Moreira, A. A., Velásquez-Orellana, O. J., Zambrano-Murillo, D. J., & Zambrano-Villamil, M. E. (2023). Importance of feedback in the student evaluation process. *International Journal of Social Sciences*, 6(3), 168–172. <https://doi.org/10.21744/ijss.v6n3.2176>
- Parker, M. J., Anderson, C., Stone, C., & Oh, Y. R. (2025). A Large Language Model Approach to Educational Survey Feedback Analysis. *International Journal of Artificial Intelligence in Education*, 35(2), 444–481. <https://doi.org/10.1007/s40593-024-00414-0>
- Prasetio, T. (2022). Umpan Balik Pada Model Pembelajaran Hybrid Mata Kuliah Ekonomi di Masa Pandemi COVID-19. *Perspektif: Jurnal Ekonomi & Manajemen Universitas Bina Sarana Informatika*. <https://doi.org/10.31294/jp.v17i1>
- Prihandini, G. R. (2022). Maukah Anda Mengerjakan Tugas Yang Lebih Sulit? Melihat Pengaruh Jenis Umpan Balik Terhadap Performa Dalam Mengerjakan Tugas (Task Performance). *ANFUSINA: Journal of Psychology*, 5(2), 213–228. <https://doi.org/10.24042/ajp.v5i2.15714>
- Razaq, M. T., Nurjanah, D., & Nurrahmi, H. (2023). Analisis Sentimen Review Film Menggunakan Naive Bayes Classifier Dengan Fitur TF-IDF. *EProceedings of Engineering*.
- Sayeed, M. S., Mohan, V., & Muthu, K. S. (2023, June 1). BERT: A Review of Applications in Sentiment Analysis. *HighTech and Innovation Journal*, Vol. 4, pp. 453–462. Ital Publication. <https://doi.org/10.28991/HIJ-2023-04-02-015>
- Shaikh, S., Daudpota, S. M., Yayilgan, S. Y., & Sindhu, S. (2024, February 5). *Exploring the potential of large-language models (LLMs) for student feedback sentiment analysis*. 214–

219. Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/fit60620.2023.00047>
- Tan, K. L., Lee, C. P., & Lim, K. M. (2023, April 1). A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. *Applied Sciences (Switzerland)*, Vol. 13. MDPI. <https://doi.org/10.3390/app13074550>
- Wahyudi, A., Gama, S., Kadek, N., & Pridayanti, Y. (2024). Evaluasi Kinerja dan Umpan Balik Untuk Pengembangan Sumber Daya Manusia di Dragonfly Konveksi. *Prosiding Seminar Nasional Pengabdian Masyarakat*, 3(1).
- Wang, Z., Xie, Q., Feng, Y., Ding, Z., Yang, Z., & Xia, R. (2024). *Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study*. Retrieved from <http://arxiv.org/abs/2304.04339>
- Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7), 5731–5780. <https://doi.org/10.1007/s10462-022-10144-1>
- Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., Li, X., Lim, Z. Y., ... Purwarianti, A. (2020). *IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding*. Retrieved from <http://arxiv.org/abs/2009.05387>
- Wu, Y., Jin, Z., Shi, C., Liang, P., & Zhan, T. (2024). Research on the application of deep learning-based BERT model in sentiment analysis. *Applied and Computational Engineering*, 71(1), 14–20. <https://doi.org/10.54254/2755-2721/71/2024MA>

LAMPIRAN