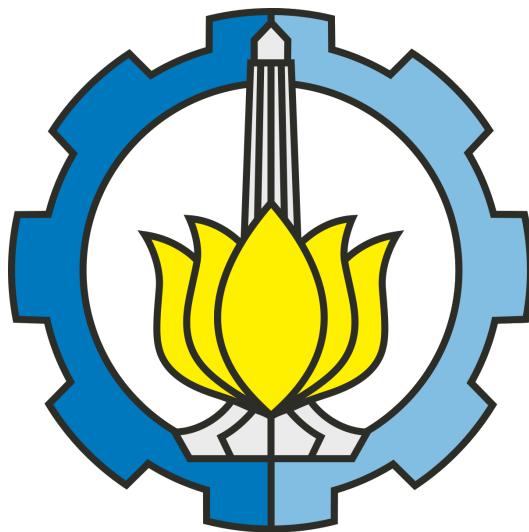


PROJECT B

ANALISIS SENTIMEN ARTIKEL BERITA GARUDA INDONESIA MENGGUNAKAN FINE-TUNING MODEL INDOBERT



Anggota Kelompok 11:
5026221005 Jason Ho
5026221144 Alfa Renaldo Aluska

PENGOLAHAN BAHASA ALAMI (A)

**DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI ELEKTRO & INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SEMESTER GASAL 2025**

DAFTAR ISI

DAFTAR ISI.....	2
ABSTRAK.....	4
BAB I	
PENDAHULUAN.....	5
1.1 Latar Belakang.....	5
1.2 Rumusan Masalah.....	6
1.3 Tujuan Penelitian.....	6
BAB II	
TINJAUAN PUSTAKA.....	7
BAB III	
METODOLOGI.....	13
3.1 Metodologi Penelitian.....	13
3.2 Uraian Alur Pelaksanaan Penelitian.....	13
3.2.1 Akuisisi Data.....	14
3.2.2 Analisis Sentimen Dasar.....	14
3.2.3 Pra-pemrosesan Data.....	14
3.2.4 Pemilihan Model.....	15
3.2.5 Fine-Tuning Model.....	16
3.2.6 Analisis Error.....	17
BAB IV	
HASIL DAN PEMBAHASAN.....	18
4.1 Hasil Akuisisi Data.....	18
4.1.1 Akuisisi Judul dan Tautan Berita.....	18
4.1.2 Tagging Berdasarkan Judul.....	21
4.1.3 Akuisisi Isi Konten Berita.....	21
4.1.4 Deskripsi Dataset.....	26
4.2 Hasil Analisis Sentimen Dasar.....	27
4.2.1 Tagging dan Klasifikasi Sentimen dengan OpenAI API.....	27
4.2.2 Verifikasi Manual Penentuan Sentimen.....	30
4.3 Hasil Praproses Data.....	30
4.3.1 Normalisasi Slang.....	31
4.3.2 Pembersihan URL.....	31
4.3.3 Pembersihan Call-to-action “Baca selanjutnya”.....	32
4.3.4 Pembersihan Encoding Artefacts.....	32
4.3.5 Pembersihan Bollerplate.....	33
4.3.6 Normalisasi Whitespace.....	33
4.3.7 Pemetaan Label Kelas.....	34
4.3.8 Pembagian Dataset.....	35
4.3.9 Tokenisasi Teks.....	36
4.3.10 Pembentukan Dataset PyTorch.....	37

4.4. Hasil Pemilihan Model.....	38
4.5 Hasil Fine-Tuning Model.....	38
4.5.1 Fungsi Loss yang Digunakan.....	39
4.5.2 Pemodelan Baseline.....	40
4.5.3 Hasil Evaluasi Model.....	42
4.5.3.1 Eksperimen 1: Penyesuaian Panjang Input Teks.....	43
4.5.3.2 Eksperimen 2: Stabilisasi Fine-Tuning dengan Warmup dan Learning Rate Scheduler.....	45
4.5.3.3 Eksperimen 3: Eksplorasi Learning Rate.....	48
4.5.3.3.1 Learning Rate = 1e-5.....	49
4.5.3.3.2 Learning Rate = 2e-5.....	50
4.5.3.3.3 Learning Rate = 3e-5.....	50
4.5.3.3.4 Perbandingan Eksperimen 3.....	51
4.5.4 Perbandingan Akhir.....	53
4.5.5 Konfigurasi Final Model yang Dipilih.....	55
4.6 Analisis Error.....	56
4.6.1 Analisis Error 1.....	56
4.6.2 Analisis Error 2.....	57
4.6.3 Analisis Error 3.....	58
4.6.4 Analisis Error 4.....	60
4.6.5 Kesimpulan Analisis Kesalahan (Error Analysis).....	61
BAB V	
PENUTUP.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	63
DAFTAR PUSTAKA.....	65
LAMPIRAN.....	67

ABSTRAK

Analisis sentimen terhadap artikel berita berbahasa Indonesia memiliki tantangan tersendiri karena karakteristik teks yang panjang, formal, serta sering mengandung sentimen implisit atau campuran. Penelitian ini bertujuan untuk melakukan analisis sentimen pada artikel berita terkait Garuda Indonesia dengan memanfaatkan model *transformer* berbasis bahasa Indonesia, yaitu IndoBERT. Dataset dikumpulkan melalui proses *web scraping* dari berbagai sumber berita daring dan dipraproses secara selektif untuk menghilangkan noise teknis tanpa menghilangkan konteks semantik teks. Proses pelabelan sentimen dilakukan menggunakan *large language model* melalui OpenAI API, kemudian diverifikasi secara manual untuk memastikan kesesuaian label dengan konteks linguistik dan makna keseluruhan berita. Model IndoBERT-base-p2 selanjutnya di-*fine-tune* untuk tugas klasifikasi sentimen tiga kelas, yaitu negatif, netral, dan positif. Eksperimen dilakukan dengan beberapa variasi konfigurasi *hyperparameter*, meliputi panjang input teks, *learning rate*, serta penerapan *learning rate scheduler*, guna mengevaluasi performa dan stabilitas pelatihan model. Hasil penelitian menunjukkan bahwa konfigurasi baseline dengan panjang input 256 token dan *learning rate* 2e-5 memberikan performa yang paling stabil dengan nilai *macro F1-score* yang tinggi. Meskipun konfigurasi lain mampu meningkatkan performa pada epoch tertentu, konfigurasi tersebut cenderung menunjukkan fluktuasi dan risiko overfitting yang lebih besar. Secara keseluruhan, hasil penelitian menunjukkan bahwa IndoBERT efektif untuk analisis sentimen artikel berita berbahasa Indonesia apabila di-*fine-tune* dengan konfigurasi yang tepat.

Kata kunci: Analisis Sentimen, Fine-Tuning, Garuda Indonesia, IndoBERT, Transformer

BAB I

PENDAHULUAN

1.1 Latar Belakang

Industri penerbangan merupakan sektor yang sangat kompetitif dan sensitif terhadap opini publik. Garuda Indonesia, sebagai maskapai nasional, secara berkelanjutan menghadapi tantangan dalam mempertahankan reputasi dan kepuasan pelanggan. Di era digital, media berita daring menjadi salah satu pembentuk utama persepsi publik. Dalam konteks ini, analisis sentimen muncul sebagai alat krusial untuk memahami opini dan emosi dari data teksual, yang terbukti membantu maskapai dalam meningkatkan kepuasan pelanggan dan memantau tren industri (Patel et al., 2022). Dengan memanfaatkan analisis sentimen, perusahaan dapat menyelaraskan layanan mereka dengan preferensi pelanggan, yang pada akhirnya akan meningkatkan loyalitas merek dan keunggulan kompetitif.

Beberapa studi terdahulu telah mengaplikasikan analisis sentimen dalam industri penerbangan menggunakan teknik *machine learning* dan *deep learning* pada data media sosial atau *review* pelanggan. Misalnya, Samir et al. (2023) menggunakan model *deep learning* pada data Skytrax, dan Li et al. (2024) melakukan analisis komparatif menggunakan BERT pada *review* pelanggan. Di Indonesia, Prasetyo et al. (2024) telah menganalisis sentimen terhadap Garuda Indonesia menggunakan data Twitter dan membandingkan model MultinomialNB, SVM, dan BERT. Namun, penelitian-penelitian tersebut didominasi oleh fokus pada data media sosial, sementara analisis sentimen dari artikel berita daring (yang memiliki karakteristik bahasa yang lebih formal dan kredibilitas tinggi terhadap opini publik) mengenai maskapai penerbangan, khususnya Garuda Indonesia, masih minim.

Dalam implementasi analisis sentimen, pemilihan metode ekstraksi fitur menjadi krusial. Teknik Term Frequency-Inverse Document Frequency (TF-IDF) merupakan metode yang efektif dan populer (Das & Chakraborty, 2019), bahkan masih kompetitif dibandingkan dengan *word embedding* modern untuk dataset terbatas (Rifaldy et al., 2025). Penggunaan TF-IDF telah terbukti memberikan hasil signifikan ketika dikombinasikan dengan algoritma *machine learning* dalam analisis sentimen (Singh et al., 2022). Selain TF-IDF, Named Entity Recognition (NER) dan Part-of-Speech (POS) tagging adalah komponen Natural Language Processing (NLP) penting yang dapat memberikan *insight* tambahan. NER yang berfungsi untuk mengidentifikasi entitas seperti nama orang, lokasi, dan organisasi (Gunawan et al., 2018) dengan model khusus telah dikembangkan untuk konteks Bahasa Indonesia (Khairunnisa et al., 2020; Wibawa et al., 2016).

Meskipun pendekatan berbasis fitur konvensional seperti TF-IDF dan identifikasi entitas (NER) mampu memberikan wawasan awal yang berharga, kompleksitas struktur bahasa dalam artikel berita menuntut metode yang lebih mampu menangkap konteks semantik secara mendalam. Oleh karena itu, penelitian ini mengusulkan pendekatan berbasis *Deep Learning* menggunakan arsitektur BERT (*Bidirectional Encoder Representations from Transformers*) yang dilakukan melalui proses *fine-tuning*. Secara spesifik, penelitian ini dirancang untuk menginvestigasi konfigurasi pelatihan yang optimal melalui serangkaian skenario eksperimen, meliputi penyesuaian batas panjang input teks (*sequence length*), stabilisasi pelatihan menggunakan *warmup* dan *scheduler*, serta eksplorasi nilai *learning rate*. Pendekatan eksperimental ini bertujuan untuk menghasilkan model klasifikasi sentimen yang

tidak hanya akurat, tetapi juga tangguh (*robust*) dalam memahami dinamika opini publik terhadap Garuda Indonesia pada media daring.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, permasalahan yang akan dikaji dalam penelitian ini adalah sebagai berikut:

1. Bagaimana kinerja model *pre-trained* BERT (*IndoBERT*) dalam mengklasifikasikan sentimen artikel berita daring berbahasa Indonesia terkait maskapai Garuda Indonesia?
2. Bagaimana pengaruh penyesuaian panjang input teks (*max sequence length*) terhadap kemampuan model dalam menangkap konteks berita dan akurasi klasifikasi, mengingat karakteristik artikel berita yang lebih panjang dibandingkan media sosial?
3. Bagaimana dampak penerapan strategi *fine-tuning* yang meliputi penggunaan *warmup*, *learning rate scheduler*, dan eksplorasi nilai *learning rate* terhadap stabilitas dan performa akhir model?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mengevaluasi kinerja model IndoBERT yang telah di-*fine-tune* untuk tugas analisis sentimen pada domain artikel berita daring mengenai Garuda Indonesia.
2. Menganalisis pengaruh batasan panjang input teks terhadap efektivitas model dalam memproses informasi kontekstual pada artikel berita.
3. Menentukan konfigurasi *hyperparameter* yang optimal, khususnya terkait strategi *learning rate* dan skema *scheduler*, untuk menghasilkan model klasifikasi sentimen yang stabil dan akurat.

BAB II

TINJAUAN PUSTAKA

Dari tinjauan atas penelitian-penelitian yang telah dilakukan sebelumnya, berikut disajikan hasil dari penelitian terdahulu dalam Tabel 2.1 untuk mempermudah pemahaman mengenai metode, data, serta temuan utama dari setiap penelitian yang relevan.

Tabel 2.1 Hasil tinjauan pustaka dari penelitian terkait

No.	Penelitian	Hasil penelitian
1	Pravina et al. (2019) Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)	Penelitian oleh Pravina, Cholissodin, & Adikara (2019) dari Universitas Brawijaya berjudul " <i>Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)</i> " bertujuan untuk mengklasifikasikan opini publik terhadap maskapai penerbangan di Indonesia melalui media sosial Twitter menjadi dua kelas sentimen: positif dan negatif. Fokus penelitian diarahkan pada lima maskapai utama, yaitu Garuda Indonesia, Citilink, Batik Air, Sriwijaya Air, dan Lion Air, yang secara aktif menggunakan Twitter untuk berinteraksi dengan pelanggan. Data diolah menggunakan metode <i>text preprocessing</i> (cleaning, stemming, translation, tokenizing, dan stopword removal) serta pembobotan kata dengan TF-IDF dan fitur Lexicon Based, sebelum akhirnya diklasifikasikan menggunakan algoritma Support Vector Machine (SVM). Hasil penelitian menunjukkan bahwa kombinasi SVM dengan fitur Lexicon Based menghasilkan performa terbaik dengan parameter optimal $C = 10$ dan <i>learning rate</i> = 0,03, menghasilkan akurasi 40%, precision 40%, recall 100%, dan F-measure 57,14%. Meskipun akurasinya tergolong rendah, penelitian ini menekankan bahwa implementasi fitur leksikon dapat meningkatkan stabilitas hasil klasifikasi, terutama pada dataset opini berbahasa campuran Indonesia–Inggris. Temuan ini memperkuat pentingnya analisis sentimen sebagai alat bantu evaluasi kualitas layanan dan reputasi maskapai seperti Garuda Indonesia, serta menunjukkan bahwa penggunaan SVM tetap relevan dalam menganalisis opini publik di sektor transportasi udara
2	Telaumbanua (2023) Analisis Sentimen Berbasis Aspek terhadap Maskapai Garuda Indonesia	Penelitian yang dilakukan oleh Telaumbanua (2023) bertujuan untuk melakukan analisis sentimen berbasis aspek terhadap opini masyarakat mengenai maskapai Garuda Indonesia di media sosial Twitter. Penelitian ini menggunakan dua algoritma utama: Latent Dirichlet Allocation (LDA) untuk mengelompokkan opini ke dalam

	melalui Media Sosial Twitter Menggunakan Algoritma Latent Dirichlet Allocation dan Multinomial Naïve Bayes	<p>aspek-aspek tertentu dan Multinomial Naïve Bayes (MNB) untuk mengklasifikasikan sentimen (positif, negatif, atau netral) pada setiap aspek.</p> <p>Dalam penelitian ini, data yang digunakan adalah 1.750 tweet yang kemudian dibagi menjadi 1.400 data latih dan 350 data uji dengan rasio 80:20. Data teks tersebut melalui beberapa tahap pra-pemrosesan yang meliputi <i>cleaning</i>, <i>case folding</i>, normalisasi, <i>tokenizing</i>, <i>filtering</i>, dan <i>stemming</i>.</p> <p>Dari hasil pemodelan topik menggunakan algoritma LDA, ditemukan tiga aspek utama yang dibicarakan oleh masyarakat, yaitu fasilitas, pelayanan, dan jadwal. Setelah aspek diidentifikasi, klasifikasi sentimen menggunakan MNB dilakukan. Evaluasi model dilakukan dengan menggunakan <i>confusion matrix</i> dan menghasilkan nilai akurasi sebesar 80%.</p>
3	<p>Daryanti & Widodo (2024)</p> <p>Analisis Sentimen Ulasan Penumpang Maskapai Penerbangan Indonesia Menggunakan Support Vector Machine, Naive Bayes, dan Random Forest</p>	<p>Penelitian Daryanti & Widodo (2024) membahas analisis sentimen ulasan penumpang maskapai penerbangan di Indonesia dengan fokus pada maskapai besar seperti Garuda Indonesia, Lion Air, Batik Air, Citilink, dan AirAsia. Latar belakang penelitian ini didasari oleh kebutuhan industri penerbangan untuk memahami persepsi publik terhadap kualitas layanan, kenyamanan, ketepatan waktu, dan profesionalitas awak kabin. Berdasarkan survei CNN Indonesia (2024), Garuda Indonesia dinyatakan sebagai maskapai paling disukai oleh masyarakat Indonesia, menegaskan posisinya sebagai acuan utama dalam evaluasi layanan penerbangan nasional</p> <p>Dalam penelitian tersebut, penulis menggunakan total 6.469 data ulasan dari platform X (Twitter) dan TripAdvisor yang diklasifikasikan menjadi tiga kategori sentimen: positif, negatif, dan netral. Proses pengolahan data dilakukan melalui tahapan <i>cleaning</i>, <i>case folding</i>, <i>tokenizing</i>, <i>stopword removal</i>, dan <i>stemming</i>, kemudian dilakukan pembobotan menggunakan metode TF-IDF. Tiga algoritma <i>machine learning</i> yang digunakan adalah Support Vector Machine (SVM), Naive Bayes, dan Random Forest, dengan evaluasi kinerja berdasarkan metrik akurasi, presisi, recall, dan F1-score.</p> <p>Hasil penelitian menunjukkan bahwa algoritma Random Forest memberikan performa terbaik dengan akurasi 91%, disusul SVM dengan 89% dan Naive Bayes 79%. Maskapai dengan sentimen positif tertinggi adalah Super Jet Air, sedangkan Garuda Indonesia tetap menjadi salah satu yang paling banyak dibahas dan dijadikan tolok ukur dalam ulasan publik. Penelitian ini menegaskan pentingnya penerapan analisis sentimen berbasis data dalam industri transportasi udara untuk membantu maskapai seperti</p>

		Garuda Indonesia memahami opini pelanggan, meningkatkan kualitas layanan, serta memperkuat reputasi merek di tengah persaingan industri penerbangan nasional
4	Prasetyo et al. (2024) A Comparative Analysis of MultinomialNB, SVM, and BERT on Garuda Indonesia Twitter Sentiment	Penelitian yang dilakukan oleh Prasetyo, Al-Majid, dan Suharjito (2024) melakukan analisis perbandingan untuk mengevaluasi kinerja model <i>machine learning</i> dalam analisis sentimen terhadap maskapai Garuda Indonesia. Studi ini menggunakan 1.250 tweet yang dikumpulkan dari Maret 2007 hingga Juli 2024 dan membandingkan tiga model berbeda: Multinomial Naïve Bayes (MultinomialNB), Support Vector Machine (SVM), dan <i>Bidirectional Encoder Representations from Transformers</i> (BERT). Untuk mengatasi masalah distribusi kelas sentimen yang tidak seimbang dalam dataset, penelitian ini menerapkan teknik <i>Synthetic Minority Over-sampling Technique</i> (SMOTE). Hasil evaluasi menunjukkan bahwa model BERT mencapai performa tertinggi dengan akurasi 75,6%, secara signifikan mengungguli SVM (71,6%) dan MultinomialNB (64,8%). Keunggulan ini disebabkan oleh kemampuan superior BERT dalam menangkap makna kontekstual dalam ulasan pelanggan, sehingga penelitian ini menyimpulkan bahwa model <i>deep learning</i> seperti BERT lebih efektif untuk tugas analisis sentimen pada data media sosial dibandingkan model <i>machine learning</i> klasik.
5	Rozi et al. (2019) Analisis Sentimen pada Opini Pengguna Maskapai Penerbangan Menggunakan Hybrid Cuckoo Search	Penelitian oleh Rozi, Arianto, & Hapsari (2019) dari Institut Teknologi Adhi Tama Surabaya berjudul “ <i>Analisis Sentimen pada Opini Pengguna Maskapai Penerbangan Menggunakan Hybrid Cuckoo Search (HCS)</i> ” berfokus pada analisis opini penumpang terhadap kualitas layanan maskapai penerbangan di Indonesia. Penelitian ini dilatarbelakangi oleh meningkatnya mobilitas masyarakat dan kebutuhan untuk menilai reputasi maskapai melalui opini daring di platform seperti Skytrax, yang memuat ulasan tentang aspek keselamatan, kenyamanan, dan pelayanan maskapai termasuk Garuda Indonesia, Lion Air, AirAsia, dan Singapore Airlines. Metode yang digunakan adalah Hybrid Cuckoo Search (HCS), yaitu gabungan antara algoritma K-Means dan Cuckoo Search untuk mengelompokkan data opini menjadi kategori positif dan negatif. Dataset berisi 3.882 ulasan penumpang dari tujuh maskapai, di antaranya Garuda Indonesia (623 ulasan). Melalui pengujian hingga 1.000 iterasi, hasil terbaik menunjukkan rata-rata akurasi 69,24%, precision 70,88%, dan recall 77,57%. Khusus untuk Garuda Indonesia, HCS menghasilkan akurasi tinggi hingga 86,19% pada iterasi ke-800. Studi ini menunjukkan bahwa HCS efektif untuk mengelompokkan opini penumpang secara cepat dan

		akurat, membantu konsumen menilai reputasi maskapai serta memberi wawasan bagi perusahaan dalam meningkatkan layanan. Dengan demikian, pendekatan ini relevan sebagai dasar pengembangan sistem analisis sentimen berbasis metaheuristik di sektor transportasi udara nasional, termasuk untuk analisis reputasi Garuda Indonesia
6	Salma (2022) Analyzing Online Public Sentiment Toward Corporate Crisis in the Age of Big Data and Automation	Penelitian Aqida Nuril Salma (2022) berjudul “ <i>Analyzing Online Public Sentiment Toward Corporate Crisis in the Age of Big Data and Automation</i> ” membahas bagaimana big data dan analisis sentimen media sosial dapat digunakan dalam manajemen krisis korporasi, dengan studi kasus pada krisis Garuda Indonesia tahun 2019. Krisis tersebut dipicu oleh ulasan negatif dari seorang vlogger tentang layanan Garuda yang kemudian viral di Twitter. Melalui pendekatan <i>text mining</i> dan <i>sentiment analysis</i> menggunakan paket TwitteR dan tm di R, penelitian ini mengumpulkan dan memvisualisasikan opini publik untuk memahami persepsi masyarakat terhadap perusahaan selama krisis. Hasil penelitian menunjukkan bahwa meskipun sempat didominasi sentimen negatif pada awal krisis, opini publik berangsur menjadi positif seiring respons cepat perusahaan dalam menangani isu. Analisis visualisasi (<i>word cloud</i> , <i>heatmap</i> , dan <i>scatterplot</i> emosi) memperlihatkan bahwa sebagian besar tweet memiliki sentimen “pleasant” atau positif, menandakan keberhasilan Garuda Indonesia memulihkan citra korporasi. Studi ini menegaskan pentingnya pemanfaatan big data dan otomasi dalam komunikasi krisis untuk membaca pola opini publik secara real time dan merancang strategi komunikasi berbasis data. Penulis juga menyoroti bahwa di era digital, praktisi humas perlu menguasai literasi data dan teknik analisis big data agar dapat beradaptasi dengan perubahan komunikasi korporasi modern. Dengan demikian, penelitian ini memberikan kontribusi penting pada kajian analisis sentimen berbasis sosial media di sektor transportasi udara, khususnya dalam konteks manajemen reputasi Garuda Indonesia selama masa krisis.
7	Seto & Septianti (2019) Dampak Kenaikan Harga Tiket Pesawat Terhadap Return dan Harga Saham pada PT. Garuda Indonesia Tbk di Bursa Efek	Sebuah studi oleh Seto dan Septianti (2019) menganalisis dampak kenaikan harga tiket pesawat terhadap <i>return</i> dan harga saham PT. Garuda Indonesia Tbk di Bursa Efek Indonesia. Penelitian komparatif ini menggunakan data mingguan selama 21 minggu sebelum dan 21 minggu sesudah kenaikan harga tiket, yang kemudian dianalisis menggunakan uji <i>paired sample t-test</i> . Hasil penelitian menunjukkan bahwa tidak terdapat perbedaan signifikan pada <i>return</i> saham sebelum dan sesudah kenaikan harga

	Indonesia	tiket, dengan nilai signifikansi sebesar 0,887. Namun, ditemukan adanya perbedaan yang signifikan pada harga saham, dengan nilai signifikansi 0,000. Harga saham rata-rata setelah kenaikan tiket pesawat tercatat lebih tinggi sebesar Rp 239,143. Perbedaan ini diatribusikan pada persepsi positif investor yang meyakini bahwa kenaikan harga akan menguntungkan perusahaan penerbangan. Kepercayaan investor yang tinggi, status PT. Garuda Indonesia Tbk sebagai BUMN, serta perbaikan sistem manajemen diyakini menjadi faktor pendorong minat investasi dan peningkatan harga saham pasca kenaikan harga tiket.
8	Sarjono et al. (2025) The Effects of COVID-19 on Garuda Indonesia Airlines	Sarjono, Alvin, dan Marcelino (2025) meneliti dampak pandemi COVID-19 terhadap maskapai Garuda Indonesia, dengan fokus pada resiliensi dan strategi adaptasi perusahaan. Analisis keuangan menunjukkan bahwa meskipun total aset meningkat karena ketergantungan pada pendanaan eksternal, Garuda Indonesia mengalami kerugian bersih yang substansial dan penurunan ekuitas. Pandemi juga menyebabkan penurunan likuiditas, yang tercermin dari rasio lancar yang menurun, serta potensi masalah solvabilitas akibat peningkatan rasio utang terhadap aset. Selain itu, terjadi penurunan pada <i>Return on Investment</i> (ROI) dan <i>Net Profit Margin</i> (NPM), yang menandakan adanya tantangan dalam profitabilitas dan efisiensi biaya. Sebagai respons, Garuda Indonesia menerapkan strategi restrukturisasi armada untuk efisiensi operasional dan berinvestasi pada layanan yang berfokus pada penumpang untuk meningkatkan pengalaman pelanggan. Penelitian ini juga menyoroti peran penting dukungan pemerintah dalam membantu industri penerbangan selama krisis.

Maskapai Garuda Indonesia, sebagai entitas bisnis vital dan BUMN, terus menjadi subjek analisis yang relevan, terutama terkait resiliensi pasca-pandemi dan dinamika pasar sahamnya (Sarjono et al., 2025; Seto & Septianti, 2019). Sejalan dengan itu, evaluasi reputasi perusahaan melalui analisis sentimen telah menjadi fokus utama dalam berbagai penelitian sebelumnya. Studi-studi tersebut telah berhasil menerapkan beragam algoritma untuk mengklasifikasikan opini dari media sosial, mulai dari *machine learning* klasik hingga model *deep learning* seperti BERT (Prasetyo et al., 2024; Daryanti & Widodo, 2024), bahkan hingga mengidentifikasi aspek spesifik seperti "pelayanan" dan "fasilitas" (Telaumbanua, 2023).

Meskipun berbagai metode telah berhasil diterapkan, mayoritas studi yang telah dilakukan tersebut masih terpaku pada pemanfaatan data media sosial dan ulasan singkat yang memiliki karakteristik bahasa tidak baku serta konteks terbatas. Sementara itu, ranah analisis sentimen terhadap artikel berita daring yang menawarkan struktur bahasa formal dan narasi yang lebih panjang masih minim dieksplorasi, khususnya dalam konteks maskapai

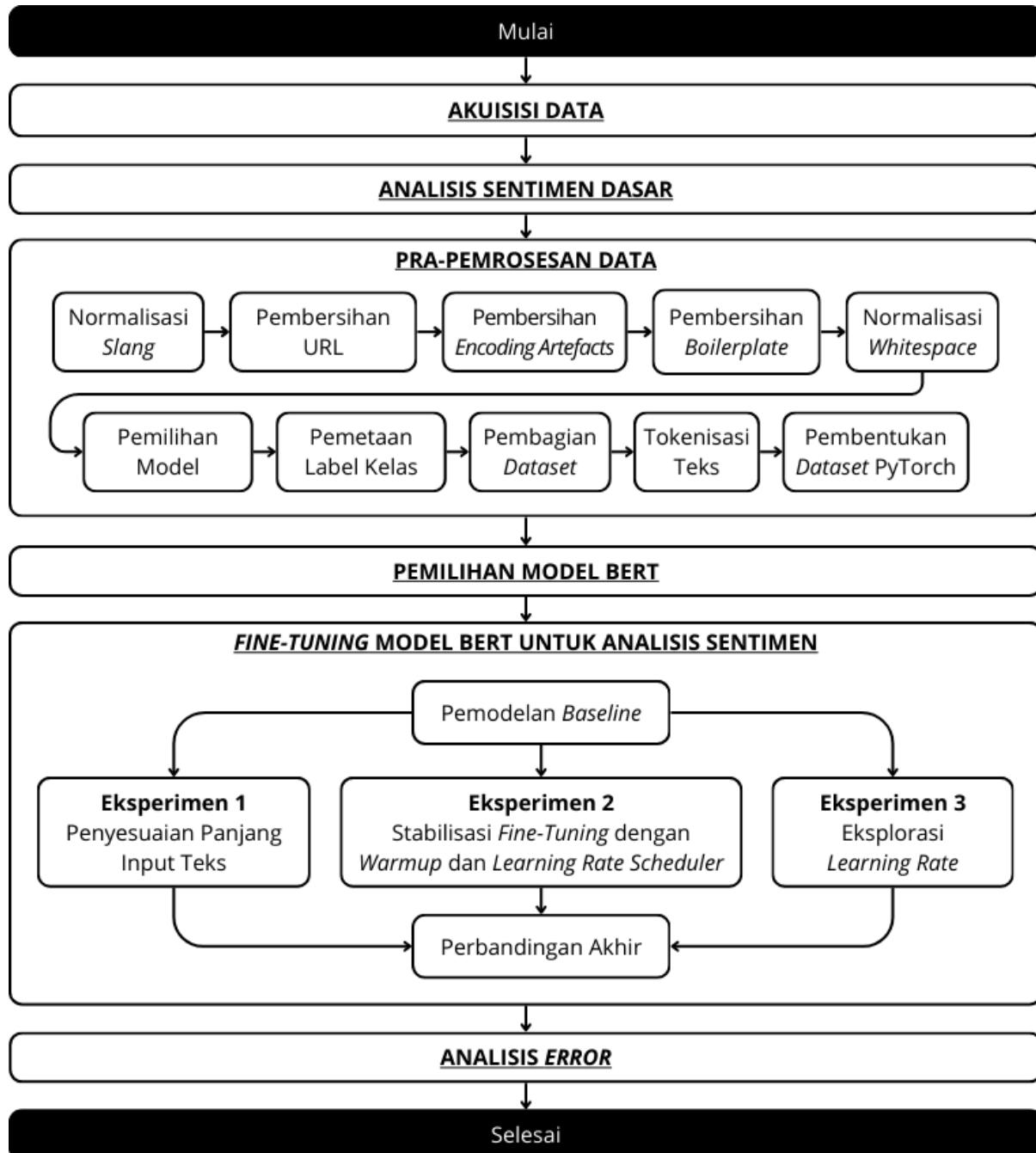
Garuda Indonesia. Padahal, penerapan model *deep learning* seperti BERT pada teks berita menuntut penanganan yang berbeda dibandingkan teks pendek media sosial, terutama terkait batasan panjang input atau *sequence length*. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan menerapkan pendekatan *Fine-Tuning* BERT pada *dataset* artikel berita daring, sekaligus melakukan eksperimen teknis mendalam terkait pengaruh panjang input teks dan optimalisasi *hyperparameter* guna menghasilkan model yang adaptif terhadap kompleksitas narasi berita.

BAB III

METODOLOGI

3.1 Metodologi Penelitian

Diagram metode penelitian yang menunjukkan langkah-langkah penggerjaan penelitian ini digambarkan pada gambar berikut ini.



Gambar 3.1 Diagram metode penelitian

3.2 Uraian Alur Pelaksanaan Penelitian

Secara garis besar, metodologi penelitian ini terdiri dari fase persiapan data, pemrosesan data, dan analisis hasil. Mengacu pada diagram alir penelitian (Gambar 3.1),

tahapan-tahapan tersebut saling berkaitan dengan luaran (*output*) dari satu tahap menjadi masukan (*input*) bagi tahap berikutnya. Proses ini diawali dengan pemanfaatan data, yang kemudian diproses lebih lanjut sebagai *dataset* utama dalam pelatihan model bahasa BERT untuk analisis sentimen. Uraian lengkap mengenai mekanisme dan prosedur di setiap tahapan disajikan dalam subbab berikut ini.

3.2.1 Akuisisi Data

Di tahap awal ini, tujuannya adalah menggunakan kembali *dataset* yang telah dikumpulkan dari bahan mentah berita dari sumber internet pada Project A (ETS). Tahapan ini mencakup:

- **Akuisisi Judul & Link Berita:** Langkah pertama dimulai dengan *crawling* atau *scraping* halaman depan portal berita untuk mengambil daftar judul beserta tautan (URL)-nya.
- **Akuisisi Data Konten Berita:** Dari tautan yang sudah didapat tadi, sistem akan membuka satu per satu halaman tersebut untuk mengambil isi berita secara utuh (teks lengkap).

3.2.2 Analisis Sentimen Dasar

<<Soal No.1: Jelaskan bagaimana Anda akan mengklasifikasikan sentimen secara MANUAL dari setiap baris data pada dataset yang telah dikumpulkan sebelumnya. Sebutkan juga alasan linguistik atau konteks yang mendukung penentuan sentimen tersebut.>>

Tahapan ini bertujuan untuk membangun *ground truth* atau data acuan yang valid pada dataset yang telah dikumpulkan. Mengingat dataset awal berupa data mentah tanpa label, proses klasifikasi sentimen akan dilakukan secara semi-manual (*human annotation*) oleh peneliti dengan meninjau setiap baris data.

Proses penentuan label didasarkan pada analisis linguistik dan konteks berita dengan kriteria klasifikasi sebagai berikut:

- **Sentimen Positif:** Label ini akan diberikan pada teks berita yang mengandung kosakata apresiasi, pencapaian prestasi, keuntungan perusahaan, atau dukungan positif.
- **Sentimen Negatif:** Label ini akan diberikan pada teks yang memuat kritik, keluhan pelanggan, kerugian finansial, atau permasalahan hukum yang dihadapi subjek berita.
- **Sentimen Netral:** Label ini akan diberikan pada teks yang bersifat informatif, faktual, berupa pengumuman, dan tidak memuat emosi subjektif yang dominan.

3.2.3 Pra-pemrosesan Data

<<Soal No.2: Untuk analisis sentimen menggunakan model pretrained BERT, proses pre-processing yang dibutuhkan adalah berbeda dari pendekatan machine learning tradisional (SVM, Naive Bayes, dll.). BERT tidak membutuhkan pre-processing agresif karena ia dilatih dengan teks mentah. Uraikan apa langkah-langkah pra-pemrosesan teks yang paling tepat untuk dataset Anda jika akan digunakan sebagai data latih untuk model analisis sentimen berbasis model BERT. Jelaskan perbedaannya dengan model BERT vs pendekatan machine learning tradisional.>>

Tahap pra-pemrosesan data dirancang untuk menyesuaikan format data mentah agar kompatibel dengan arsitektur model *Transformer* (BERT). Prosedur ini memiliki perbedaan mendasar dengan pendekatan *Machine Learning* tradisional (seperti SVM atau Naive Bayes). Jika metode tradisional umumnya memerlukan pembersihan agresif seperti *stemming* dan *stopword removal*, penelitian ini tidak akan menerapkan langkah tersebut. Hal ini dikarenakan BERT membutuhkan struktur kalimat yang utuh, termasuk kata sambung dan imbuhan, untuk dapat mempelajari konteks semantik secara mendalam. Langkah-langkah pra-pemrosesan yang akan dilakukan meliputi:

- **Normalisasi Slang:** Kata-kata tidak baku, singkatan, atau bahasa gaul diganti menjadi bentuk formalnya menggunakan kamus leksikon kustom agar sesuai dengan kosakata yang dipelajari model.
- **Pembersihan URL:** Tautan web atau *hyperlink* yang ada di dalam teks dihapus menggunakan pola Regex karena tidak memiliki nilai sentimen.
- **Pembersihan Encoding Artefacts:** Karakter yang rusak atau simbol aneh akibat kesalahan *decoding* (seperti ™) diperbaiki menggunakan pustaka ftfy agar tidak menjadi token yang tidak dikenali (*unknown token*).
- **Pembersihan Boilerplate:** Teks navigasi situs berita yang tidak relevan dengan konten (seperti "Baca selengkapnya", kredit editor, atau menu *footer*) dihapus agar model fokus pada inti berita.
- **Normalisasi Whitespace:** Spasi ganda, baris kosong berlebih, dan spasi di awal/akhir kalimat dirapikan menjadi satu spasi tunggal untuk efisiensi tokenisasi.
- **Pemetaan Label Kelas:** Label kategori sentimen teks dikonversi menjadi format numerik (Negatif=0, Netral=1, Positif=2) agar dapat diproses oleh fungsi *loss* pada model.
- **Pembagian Dataset:** Data dibagi menggunakan teknik *stratified splitting* menjadi data latih (80%), data validasi (10%), dan data uji (10%) untuk memastikan evaluasi yang objektif.
- **Tokenisasi Teks:** Kalimat diubah menjadi urutan *token ID* dan *attention mask* menggunakan *tokenizer* bawaan IndoBERT, dengan batas panjang input (*max length*) yang ditentukan.
- **Pembentukan Dataset PyTorch:** Hasil tokenisasi dan label dibungkus ke dalam objek kelas Dataset PyTorch agar dapat dimuat secara efisien dalam *batch* selama proses pelatihan.

3.2.4 Pemilihan Model

<<Soal No.3: Model BERT apa yang paling tepat digunakan melakukan analisis sentimen untuk dataset Anda? Sertakan alasan terkait ukuran model, domain bahasa, dan trade-off performa terhadap kecepatan.>>

Tahap ini berfokus pada penentuan arsitektur model yang akan digunakan sebagai landasan (*backbone*) sistem. Pemilihan varian model BERT didasarkan pada analisis terhadap tiga kriteria metodologis utama:

- **Domain Bahasa:** Penelitian ini akan memprioritaskan model yang telah dilatih (*pre-trained*) pada korpus Bahasa Indonesia (IndoBERT) dibandingkan model multibahasa, guna menjamin representasi semantik yang lebih akurat untuk teks berita nasional.
- **Ukuran Model:** Pemilihan antara varian *Base* dan *Large* akan mempertimbangkan ketersediaan sumber daya komputasi. Varian *Base* diproyeksikan sebagai pilihan utama karena memiliki efisiensi memori yang lebih baik.
- **Trade-off Performa terhadap Kecepatan:** Model yang dipilih harus menawarkan keseimbangan optimal antara akurasi prediksi dengan waktu komputasi (*training* dan *inference time*).

3.2.5 Fine-Tuning Model

<<Soal No.4: Jelaskan langkah-langkah utama melakukan fine-tuning BERT untuk kelas sentimen dataset Anda (misal: positif, negatif, netral) menggunakan HuggingFace Transformers. Minimal jelaskan: Tokenisasi, Loss function yang digunakan, Evaluasi model.>>

Pada tahap ini, model *pre-trained* yang telah dipilih akan dilatih ulang (*fine-tuning*) untuk tugas spesifik klasifikasi sentimen. Proses ini akan diimplementasikan menggunakan kerangka kerja *HuggingFace Transformers* dengan penjelasan:

- **Persiapan Input:** Dataset yang telah ditokenisasi akan dikonversi menjadi objek Dataset PyTorch agar dapat diproses dalam *batch*.
- **Fungsi Loss:** Pelatihan akan menggunakan Cross-Entropy Loss sebagai fungsi objektif untuk mengukur dan meminimalkan selisih antara probabilitas prediksi model dengan label sentimen sebenarnya.
- **Evaluasi Model:** Selama proses pelatihan, evaluasi akan dilakukan secara berkala pada setiap *epoch* menggunakan data validasi. Metrik yang digunakan adalah F1-Score untuk memantau performa model dan menentukan titik henti terbaik guna menghindari *overfitting*.

Proses ini dilakukan melalui beberapa skenario eksperimen untuk mencari konfigurasi terbaik. Tahapan ini mencakup:

- **Pemodelan Baseline:** Eksperimen awal menggunakan pengaturan standar (Learning Rate 2e-5, Max Length 256) untuk mendapatkan tolok ukur performa dasar model.
- **Eksperimen 1 (Penyesuaian Panjang Input):** Batas panjang teks dinaikkan menjadi 512 token untuk melihat apakah konteks yang lebih panjang dapat meningkatkan akurasi, meski berisiko *overfitting*.
- **Eksperimen 2 (Stabilisasi Pelatihan):** Penerapan teknik *warmup* dan *learning rate scheduler* untuk membuat proses *update* bobot model di awal pelatihan menjadi lebih halus dan stabil.
- **Eksperimen 3 (Eksplorasi Learning Rate):** Pengujian variasi nilai *learning rate* (1e-5, 2e-5, 3e-5) untuk mencari keseimbangan terbaik antara kecepatan belajar model dan kestabilan konvergensi.

- **Perbandingan Akhir:** Hasil dari seluruh skenario dievaluasi berdasarkan F1-Score, di mana konfigurasi *Baseline* akhirnya dipilih sebagai model final karena memberikan performa paling stabil dan efisien.

3.2.6 Analisis Error

<<Soal No.5: Buat sejumlah contoh prediksi yang SALAH (*misclassification*) dari model BERT Anda dan jelaskan mengapa model bisa keliru. Diskusikan apakah kesalahan tersebut bersifat: lexical ambiguity, domain mismatch, kurang konteks, kalimat campuran (*mixed sentiment*), dsb.>>

Setelah model final terbentuk, dilakukan analisis terhadap data uji yang gagal diprediksi dengan benar (*misclassification*) untuk memahami kelemahan model. Analisis ini bertujuan untuk mengidentifikasi penyebab kegagalan model yang dalam beberapa kasus bersifat:

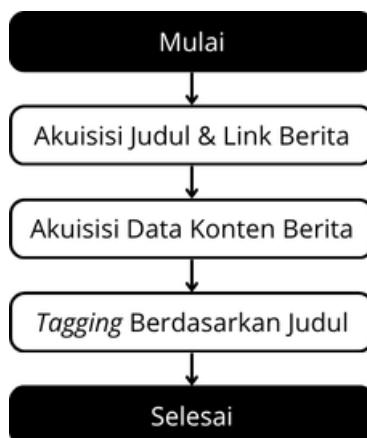
- **Lexical Ambiguity (Ambiguitas Leksikal):** Kesalahan akibat kata yang memiliki makna ganda bergantung pada konteks kalimat (misalnya kata "jatuh" dalam konteks harga vs kecelakaan).
- **Mixed Sentiment (Sentimen Campuran):** Kesalahan pada kalimat yang memuat dua polaritas sentimen berlawanan sekaligus, sehingga menyulitkan model menentukan sentimen dominan.
- **Lack of Context (Kurang Konteks):** Kesalahan akibat informasi kunci yang hilang karena pemotongan teks atau kalimat yang memerlukan pengetahuan implisit (*world knowledge*).
- **Domain Mismatch:** Kesalahan akibat munculnya istilah teknis atau jargon spesifik yang jarang ditemui model selama masa pra-pelatihan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Akuisisi Data

Tahapan akuisisi data merupakan proses awal dalam penelitian ini yang bertujuan untuk mengumpulkan, menyiapkan, dan mengorganisasi data berita yang akan diproses dan dianalisis lebih lanjut. Proses ini dilakukan secara sistematis melalui beberapa langkah berurutan agar data yang diperoleh relevan, terstruktur, dan siap digunakan pada tahap pra-pemrosesan dan analisis. Secara umum, alur akuisisi data dimulai dari pengambilan judul dan tautan berita dari sumber daring resmi, kemudian dilanjutkan dengan pengambilan isi berita (konten) berdasarkan tautan yang telah diperoleh. Setelah seluruh berita terkumpul, dilakukan penandaan awal (*tagging*) untuk memberi label kategori pada setiap berita, sesuai dengan konteks penelitian yang telah ditetapkan. Urutan lengkap tahapan akuisisi data dapat dilihat pada diagram berikut.



Gambar 4.1 Diagram Alir Tahap Scraping Berita

4.1.1 Akuisisi Judul dan Tautan Berita

Pada tahap ini dilakukan proses pengambilan (*scraping*) data berita dari Google News menggunakan bahasa pemrograman Python dengan bantuan pustaka Selenium. Langkah awal mencakup instalasi berbagai dependensi penting seperti *wget*, *curl*, *unzip*, serta pustaka Python yaitu Selenium, Chromedriver-Autoinstaller, dan Dateparser. Selain itu, Google Chrome dan Chromedriver juga diinstal agar proses otomasi peramban dapat berjalan dalam mode headless (tanpa antarmuka grafis). Artikel yang diambil akan terbatas hingga tanggal 30 September 2025, dengan total artikel yang berhasil diambil adalah 1194 artikel. Dataset yang berhasil diambil, dilakukan data pre-processing dan siap untuk dianalisis dijelaskan di Tabel 3.1 berikut.

Tabel 4.1 Jumlah artikel dalam *dataset* per tahapan

No.		Tahapan	Dihapus	Jumlah Akhir
1		Akuisisi judul dan tautan berita	-0	1194

2	Hanya judul dan tautan berita	Hapus baris <i>scraping</i> judul dan tautan berita kosong karena gagal di- <i>scrap</i>	-191	1003
3		Filter bahasa Inggris judul berita dengan formula Google Spreadsheets, lalu dihapus dengan Python	-389	614
4		<i>Scraping</i> isi konten berdasarkan tautan yang telah di- <i>scrap</i> sebelumnya,	-0	614
5	Beserta dengan isi konten berita	Penghapusan baris hasil <i>scraping</i> isi konten yang kosong	-81	533
6		Penghapusan baris hasil <i>scraping</i> isi konten berbahasa Inggris dengan fungsi <code>detect_language()</code>	-38	495
7		Filter dan hapus konten berbahasa Inggris manual melalui Microsoft Excel	-26	469

Setelah seluruh dependensi terpasang, dilakukan konfigurasi opsi Chrome agar dapat dijalankan secara aman di lingkungan kerja. Selanjutnya, dibuat struktur data berupa *set* untuk menyimpan hasil scraping agar tidak terjadi duplikasi tautan.

Fungsi utama yang digunakan adalah `scrape_google_news_link()`. Fungsi ini bertugas untuk mengakses hasil pencarian berita di Google News berdasarkan kata kunci tertentu, yaitu “Garuda Indonesia” dan “Pesawat Garuda”. Melalui Selenium, *browser* otomatis membuka halaman-halaman hasil pencarian, kemudian mengambil sejumlah artikel per batch (setiap 10 artikel), dan mengekstrak beberapa komponen penting dari masing-masing berita, meliputi: tautan berita (*link*), judul artikel, tanggal publikasi, dan nama portal berita. Data yang telah dikumpulkan disimpan dalam variabel `all_articles` yang berisi kumpulan data berita unik.

```
Mengakses URL: https://www.google.com/search?q=garuda+indonesia
Halaman artikel sudah tidak tersedia
Proses selesai. Total link artikel unik yang berhasil diambil: 1194
-----
Mengakses URL: https://www.google.com/search?q=pesawat+garuda
Halaman artikel sudah tidak tersedia
Proses selesai. Total link artikel unik yang berhasil diambil: 1194
```

Gambar 4.2 Hasil proses scraping

Setelah proses pengumpulan selesai, sebagian data ditampilkan ke layar sebagai contoh untuk memastikan hasil *scraping* berjalan dengan baik. Bagian ini memperlihatkan beberapa entri awal berupa tautan, judul, tanggal, dan portal berita.

```
('https://www.travelandtourworld.com/news/article/garuda-indonesia-now-expands-flight-network-to-bali-yogyakarta-surabaya-and-other-main-tourism-hub-by-2029-heres-what-you-need-to-know/', 'Garuda Indonesia Now Expands Flight Network to Bali, Yogyakarta, Surabaya, and Other Main Tourism Hub by 2029: Here's What You Need To Know', '2025-07-31', 'Travel And Tour World')
('https://www.ft.com/content/45ff1892-b4fe-4caf-9d07-2e6a0a189fed', 'Islamic bonds come under microscope after Garuda Indonesia default', '2021-08-17', 'Financial Times')
('https://ulasan.co/beredar-video-ban-pesawat-garuda-menggelinding-di-landasan-pacu-tanjungpinang/', 'Beredar Video Ban Pesawat Garuda Menggelinding di Landasan Pacu Tanjungpinang', '2025-04-16', 'Ulasan.co')
('https://www.cnnindonesia.com/ekonomi/20240824164025-92-1137068/135-penumpang-pesawat-tangki-bocor-garuda-sudah-diterbangkan-kembali', '135 Penumpang Pesawat Tangki Bocor Garuda Sudah Diterbangkan Kembali', '2024-08-24', 'CNN Indonesia')
('https://tirto.id/cara-pilih-kursi-garuda-terbaru-dan-besaran-tarifnya-g65a', 'Cara Pilih Kursi Garuda Terbaru dan Besaran Tarifnya', '2024-12-31', 'Tirto.id')
```

Gambar 4.3 Entri awal hasil scraping

Tahap berikutnya adalah mengonversi data hasil scraping ke dalam bentuk DataFrame menggunakan pustaka Pandas. DataFrame ini memiliki empat kolom utama yaitu *link*, *judul*, *tanggal*, dan *portal*. Kolom tanggal kemudian dikonversi ke format DateTime dan diurutkan dari tanggal terbaru ke tanggal terlama agar memudahkan proses analisis di tahap berikutnya.

	link	judul	tanggal	portal
0	https://kumparan.com/kumparanbisnis/garuda-ind...	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	2025-09-30	Kumparan
1	https://voi.id/en/economy/519004	Commission V DPR Will Investigate Allegations ...	2025-09-29	VOI.ID
2	https://www.prnewswire.com/news-releases/garud...	Garuda Indonesia Goes Digital: Air Cargo Capac...	2025-09-29	PR Newswire
3	https://in.investing.com/news/company-news/gar...	Garuda Indonesia adds air cargo capacity to We...	2025-09-29	Investing.com India
4	https://jambi.pikiran-rakyat.com/info-data/pr...	Jadwal Kedatangan Pesawat di Bandara Sultan Th...	2025-09-29	Jambian
...
1189	https://www.thejakartapost.com/indonesia/2024/...	Garuda Indonesia flight makes emergency landin...	NaT	The Jakarta Post
1190	https://finance.detik.com/bursa-dan-valas/d-81...	Gelar RUPSLB Lagi, Anak Usaha Garuda Tunjuk Di...	NaT	detikFinance
1191	https://djsaviation.net/garuda-indonesia-aircr...	Garuda Indonesia Aircraft Are Grounded	NaT	Dj's Aviation
1192	https://www.aviacionline.com/qatar-airways-and...	Qatar Airways and Garuda Indonesia expand part...	NaT	Aviacionline
1193	https://www.flightglobal.com/airlines/garuda-s...	Garuda secures key payment from Indonesian gov...	NaT	FlightGlobal

1194 rows × 4 columns

Gambar 4.4 Preview dataset hasil scraping

Hasil akhir dari proses ini disimpan ke dalam dua format file, yaitu CSV dan Excel, dengan nama *link_berita_garudaindonesia.csv* dan *link_berita_garudaindonesia.xlsx*. File tersebut berisi kumpulan tautan berita lengkap dengan judul, tanggal publikasi, serta nama portal berita yang menjadi hasil akhir tahap akuisisi.

Setelah *scraping* menggunakan Python, didapatkan data sebanyak 1004 baris, dengan 1003 artikel dan 1 *header*. Akan tetapi, pada data tersebut, terdapat beberapa judul berita dalam bahasa Inggris. Berita dalam bahasa Inggris kemudian di-*detect* melalui Google Spreadsheets dan dilabeli dengan kode 'EN'. Setelah itu, dilakukan *filtering* untuk menghapus semua baris berita berbahasa Inggris.

```
=IF(AND(LEN(TRIM(A2))>=10, DETECTLANGUAGE(A2)="en"), "EN", "KEEP")
```

Gambar 4.5 Formula deteksi bahasa Inggris

4.1.2 Tagging Berdasarkan Judul

Penentuan *tag* dilakukan berdasarkan kecocokan ekspresi reguler (*regex*) dari judul berita (setelah diubah menjadi huruf kecil) dengan daftar kata kunci yang telah ditentukan menggunakan fungsi =ARRAYFORMULA() di Google Spreadsheets.

```
=ARRAYFORMULA(  
IF(A2:A="","",  
IFS(  
REGEXMATCH(LOWER(A2:A),"^(laba|rugi|pendapatan|utang|restrukturisasi|pkpu|obligasi|kuartal|laporan keuangan)$)","Keuangan",  
REGEXMATCH(LOWER(A2:A),"^(rute|penerbangan baru|buka rute|tutup rute|frekuensi|penerbangan|operasional|insiden|delay|batal|keselamatan)$)","Rute/Operasional",  
REGEXMATCH(LOWER(A2:A),"^(direktur|komisaris|manajemen|rups|ceo|pergantian|pengurus)$)","Manajemen",  
REGEXMATCH(LOWER(A2:A),"^(kasus|pengadilan|kpk|dugaan|suap|hukum|sidang|sanksi|denda|izin)$)","Hukum/Regulasi",  
TRUE,"Lainnya"  
)  
)
```

Gambar 4.6 Formula untuk *tagging* berdasarkan judul

Keterangan:

- Tag "Keuangan" diberikan jika teks mengandung salah satu kata berikut: *laba, rugi, pendapatan, utang, restrukturisasi, pkpu, obligasi, kuartal*, atau *laporan keuangan*.
- Tag "Rute/Operasional" diberikan jika teks mengandung salah satu kata berikut: *rute, penerbangan baru, buka rute, tutup rute, frekuensi penerbangan, operasional, insiden, delay, batal*, atau *keselamatan*.
- Tag "Manajemen" diberikan jika teks mengandung salah satu kata berikut: *direktur, komisaris, manajemen, rups, ceo, pergantian, pengurus*.
- Tag "Hukum/Regulasi" diberikan jika teks mengandung salah satu kata berikut: *kasus, pengadilan, kpk, dugaan, suap, hukum, sidang, sanksi, denda, izin*.
- Tag "Lainnya" diberikan jika teks tidak memenuhi kriteria kata kunci dari salah satu kategori di atas.

Kolom *output* dari proses ini tetap ada dalam *dataset* karena penggunaannya untuk Project A (ETS), tetapi tidak digunakan sebagai acuan pelatihan model untuk Project B (EAS).

4.1.3 Akuisisi Isi Konten Berita

Tahap ini bertujuan untuk mengekstrak isi berita dari daftar tautan yang telah diperoleh pada tahap sebelumnya. Data awal berupa berkas data_link_berita.csv sudah berisi kolom judul dan tautan berita. Pada tahap ini, sistem hanya berfokus untuk mengunduh serta mengekstrak teks utama dari setiap tautan tersebut.

Proses diawali dengan instalasi beberapa pustaka pendukung seperti Pandas, Tqdm, Requests, Newspaper3k, Trafilatura, dan Readability-LXML. Pustaka-pustaka ini digunakan untuk membaca data, melakukan permintaan ke situs berita, serta mengekstrak isi utama halaman web secara otomatis.

Setiap tautan dalam *dataset* diproses satu per satu menggunakan beberapa metode ekstraksi yang disusun secara berlapis. Metode pertama adalah Newspaper3k, yang berusaha mengambil judul dan isi artikel secara langsung. Jika hasilnya terlalu pendek atau gagal, sistem beralih ke Trafilatura, yang mengekstrak teks berdasarkan struktur HTML halaman. Bila kedua metode tersebut masih belum menghasilkan teks yang memadai, digunakan Readability-LXML untuk menyaring bagian teks yang paling relevan dari halaman tersebut. Selama proses pengambilan konten, sistem menggunakan mekanisme *timeout, retry* otomatis, serta jeda acak antar-request untuk mencegah kegagalan akibat pembatasan akses (error 429).

Setiap teks hasil ekstraksi juga dibersihkan dari spasi berlebih dan disusun ulang agar tampil lebih rapi.

Setelah seluruh tautan berhasil diproses, hasil ekstraksi berupa konten berita ditempatkan bersebelahan dengan kolom judul di dalam dataframe. Data akhir kemudian disimpan kembali dalam berkas keluaran berformat .xlsx, dengan nama yang sama seperti berkas input namun ditambahkan akhiran _with_content.xlsx. Hasil akhir tahap ini adalah dataset yang telah berisi teks lengkap dari masing-masing berita, siap digunakan pada tahap pra-pemrosesan teks berikutnya.

```
Mengambil konten artikel: 100%|██████████| 614/614 [42:42<00:00,  4.17s/it]
Selesai! Tersimpan: data_link_berita_with_content.csv
```

Gambar 4.7 Hasil scraping isi konten berita

Meskipun sudah dilakukan tahapan *filtering* judul artikel berbahasa Inggris (sebagaimana dalam bagian 3.1), beberapa artikel berbahasa Inggris tetap masuk dalam *dataset*. Oleh karena itu, dilakukan tahapan untuk membersihkan dan memfilter data hasil *scraping* sebelumnya agar hanya tersisa data yang valid dan relevan untuk proses analisis berikutnya. Proses diawali dengan membaca *file* hasil akuisisi, yaitu *data_link_berita_with_content.csv*, menggunakan pustaka Pandas. File tersebut berisi data mentah berupa tautan, judul, tanggal, portal, serta konten berita yang telah berhasil dikumpulkan dari Google News.

Apabila *file* berhasil dimuat, program akan menampilkan beberapa baris pertama untuk memastikan struktur data telah sesuai. Jika file tidak ditemukan, sistem akan memberikan pesan kesalahan agar pengguna dapat memperbaiki jalur file yang digunakan.

	link	judul	konten	tanggal	portal	tag
0	https://kumparan.com/kumparanbisnis/garuda-ind...	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	2025-09-30 00:00:00	Kumparan	Manajemen
1	https://www.bloombergtechnoz.com/detail-news/8...	Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir...	Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir...	2025-09-29 00:00:00	Bloomberg Technoz	Manajemen
2	https://voi.id/ekonomi/519004/komisi-v-dpr-bak...	Komisi V DPR Bakal Dalam Dugaan Mafia Jual Be...	JAKARTA - Ketua Komisi V DPR Lasarus mengataka...	2025-09-29 00:00:00	VOI.ID	Rute/Operasional
3	https://in.investing.com/news/company-news/gar...	Garuda Indonesia adds air cargo capacity to We...	Nan	2025-09-29 00:00:00	Investing.com India	Lainnya
4	https://www.kompasiana.com/zainularifin2714/68...	Rencana Merger Garuda Indonesia - Pelita Air: ...	Latar Belakang\nPada pertengahan 2023, wacana ...	2025-09-29 00:00:00	Kompasiana.com	Lainnya

Gambar 4.8 *Preview* hasil *scraping* mentah

Langkah pertama dalam pembersihan data adalah menghapus baris yang tidak memiliki nilai pada kolom “konten”. Hal ini dilakukan menggunakan fungsi dropna() agar hanya berita yang memiliki isi lengkap yang tersisa dalam dataset. Berdasarkan hasil eksekusi, jumlah data awal sebanyak 614 baris berkurang menjadi 533 baris setelah pembersihan, yang berarti terdapat 81 data tanpa isi konten yang dihapus.

Original DataFrame shape: (614, 6)
 Cleaned DataFrame shape: (533, 6)

	link	judul	konten	tanggal	portal	tag
0	https://kumparan.com/kumparanbisnis/garuda-indonesia-kembali-rupslb-di-tengah-isu	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	2025-09-30 00:00:00	Kumparan	Manajemen
1	https://www.bloombergtechnoz.com/detail-news/8...	Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir...	Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir...	2025-09-29 00:00:00	Bloomberg Technoz	Manajemen
2	https://voi.id/ekonomi/519004/komisi-v-dpr-bakal-dalam...	Komisi V DPR Bakal Dalamai Dugaan Mafia Jual Be...	JAKARTA - Ketua Komisi V DPR Lasarus mengataka...	2025-09-29 00:00:00	VOI.ID	Rute/Operasional
4	https://www.kompasiana.com/zainularifin2714/68...	Rencana Merger Garuda Indonesia - Pelita Air ...	Latar Belakang\nPada pertengahan 2023, wacana ...	2025-09-29 00:00:00	Kompasiana.com	Lainnya
5	https://www.cnnindonesia.com/ekonomi/202509292...	Dony Oskaria Pastikan Merger Pelita Air-Garuda...	--\nPlt Menteri Badan Usaha Milik Negara (BUMN...)	2025-09-29 00:00:00	CNN Indonesia	Lainnya

Gambar 4.9 *Preview hasil scraping* setelah penghapusan baris kosong

Selanjutnya dilakukan deteksi bahasa pada kolom “judul” menggunakan pustaka LangDetect. Fungsi khusus detect_language() dibuat untuk mengidentifikasi bahasa setiap judul berita dengan menangani kemungkinan error, misalnya jika teks kosong atau bukan bertipe string. Setelah bahasa terdeteksi, data yang berbahasa Inggris (kode “en”) dihapus agar hanya berita berbahasa Indonesia yang dipertahankan dalam dataset.

df_cleaned['detected_language'] = df_cleaned['judul'].apply(detect_language)						
	link	judul	konten	tanggal	portal	tag
0	https://kumparan.com/kumparanbisnis/garuda-indonesia-kembali-rupslb-di-tengah-isu	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	Garuda Indonesia Kembali RUPSLB di Tengah Isu ...	2025-09-30 00:00:00	Kumparan	Manajemen
1	https://www.bloombergtechnoz.com/detail-news/8...	Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir...	Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir...	2025-09-29 00:00:00	Bloomberg Technoz	Manajemen
2	https://voi.id/ekonomi/519004/komisi-v-dpr-bakal-dalam...	Komisi V DPR Bakal Dalamai Dugaan Mafia Jual Be...	JAKARTA - Ketua Komisi V DPR Lasarus mengataka...	2025-09-29 00:00:00	VOI.ID	Rute/Operasional
4	https://www.kompasiana.com/zainularifin2714/68...	Rencana Merger Garuda Indonesia - Pelita Air: ...	Latar Belakang\nPada pertengahan 2023, wacana ...	2025-09-29 00:00:00	Kompasiana.com	Lainnya
5	https://www.cnnindonesia.com/ekonomi/202509292...	Dony Oskaria Pastikan Merger Pelita Air-...	--\nPlt Menteri Badan Usaha Milik Negara (BUMN...)	2025-09-29 00:00:00	CNN Indonesia	Lainnya

Gambar 4.10 *Preview dataset* setelah penghapusan artikel berbahasa Inggris

Data berita yang berbahasa Inggris kemudian disimpan dalam *dataframe* terpisah. Tujuannya adalah untuk menampilkan daftar berita yang terdeteksi menggunakan bahasa Inggris untuk diperiksa sebelum dihapus dari *dataset* utama.

Rows removed (English titles):							
		link	judul	konten	tanggal	portal	tag detected_languag
19	https://www.ch-aviation.com/news/158639-citilink-garuda-indonesia-to-reactivate-more...	Citilink, Garuda Indonesia to reactivate more ...	Aviation Intelligence for your everyday use\nO...	2025-09-27 00:00:00	ch-aviation	Lainnya	en
33	https://www.ch-aviation.com/news/158613-garuda-indonesia-says-pelita-air-merger-in-ear...	Garuda Indonesia says Pelita Air merger in ear...	Aviation Intelligence for your everyday use\nO...	2025-09-25 00:00:00	ch-aviation	Lainnya	en
48	https://www.kompas.id/artikel/en-berat-sebelah...	The One-Sided Merger of Garuda Indonesia and P...	The issue of merging PT Garuda Indonesia (Pers...	2025-09-24 00:00:00	Kompas.id	Lainnya	en
62	https://en.tempo.co/read/2050959/garuda-indonesia-to-launch-new-halim-palembang...	Garuda Indonesia to Launch New Halim-Palembang...	TEMPO.CO, Jakarta - PT Garuda Indonesia (Perse...	2025-09-23 00:00:00	Tempo.co	Lainnya	en
66	https://avitrader.com/2025/09/19/turkish-techn...	Turkish Technic expands MRO ties with Garuda I...	At the MRO Asia Pacific 2025 in Singapore, Tur...	2025-09-23 00:00:00	AviTrader Aviation News	Lainnya	en
68	https://www.dealstreetasia.com/stories/garuda-indonesia-pelita-air-merger-plan-gets-n...	Garuda Indonesia-Pelita Air merger plan gets n...	Register Now to Unlock Premium Content\nJoin t...	2025-09-23 00:00:00	DealStreetAsia	Lainnya	en

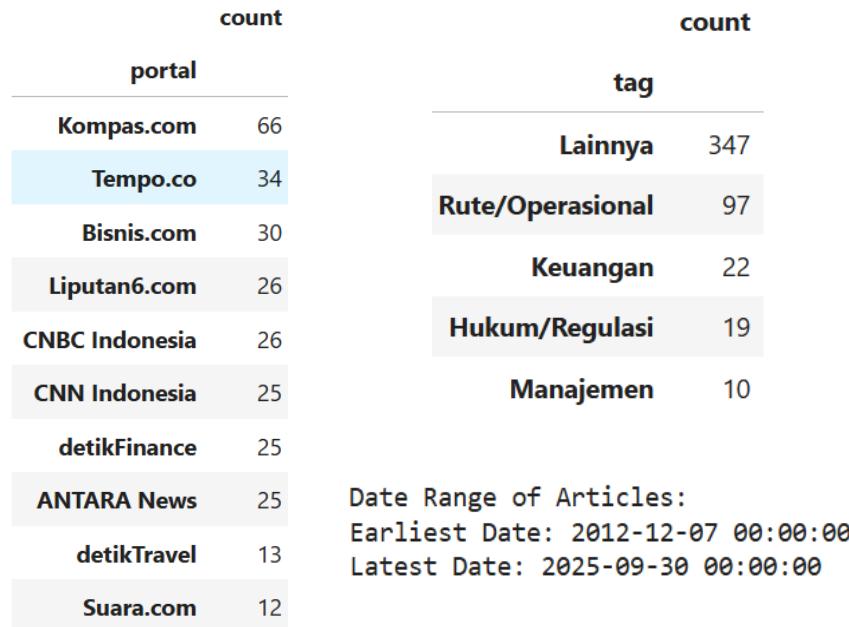
Gambar 4.11 Preview dataset berbahasa Inggris yang dihapus

Setelah proses penyaringan selesai, data bersih disimpan ke dalam file baru dengan nama *data_link_berita_with_content_cleaned.csv*. File ini berisi berita dengan konten lengkap dan berbahasa Indonesia yang siap digunakan pada tahap analisis berikutnya.

Sebagai tahap akhir, dilakukan analisis deskriptif awal terhadap dataset hasil pembersihan. Analisis ini mencakup informasi statistik dasar, jumlah nilai unik pada setiap kolom kategorikal, serta distribusi data berdasarkan portal berita dan tag yang ada. Selain itu, kolom *tanggal* dikonversi ke format waktu standar untuk melihat rentang waktu publikasi berita yang berhasil dikumpulkan. Hasil analisis menunjukkan persebaran artikel dari berbagai portal dengan periode publikasi tertentu yang akan menjadi dasar untuk analisis lanjut.

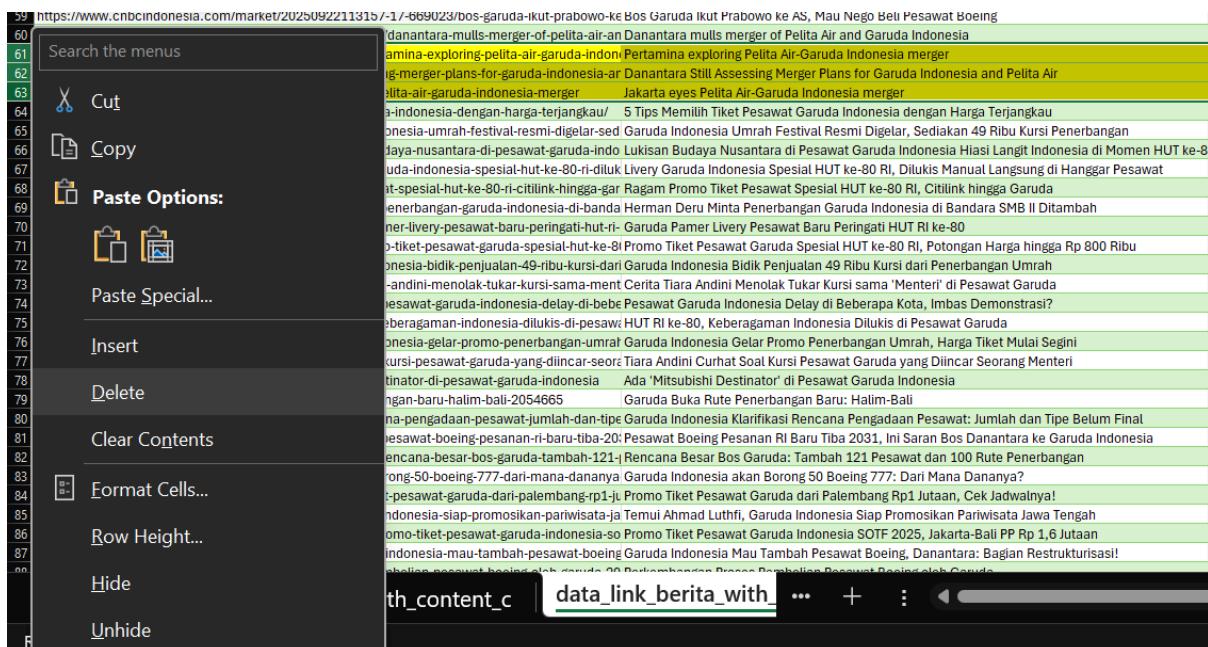
Descriptive Statistics:							
	link	judul	konten	tanggal	portal	tag	
count	495	495	495	495	495	495	495
unique	495	494	494	191	104	5	
top	https://www.tempo.co/ekonomi/tahun-depan-garud...	Danantara injects US\$405 million into Garuda I...	Aviation Intelligence for your everyday use\nO...	2025-06-10 00:00:00	Kompas.com	Lainnya	
freq	1	2	2	19	66	347	

Gambar 4.12 Distribusi hasil *scraping* konten setelah *cleaning*



Gambar 4.13 Distribusi hasil *scraping* konten setelah *cleaning* per portal dan tag

Selanjutnya, isi *file* diperiksa secara manual melalui Microsoft Excel. Dari hasil pemeriksaan, ditemukan bahwa, meskipun judul-judul bahasa Inggris sudah dihapus sebelum dilakukan akuisisi isi konten berita, beberapa judul dan konten berbahasa Inggris masih saja tetap masuk ke dalam *dataset*. Oleh karena itu, dilakukan pembersihan secara manual dengan penandaan dan penghapusan dalam Microsoft Excel.

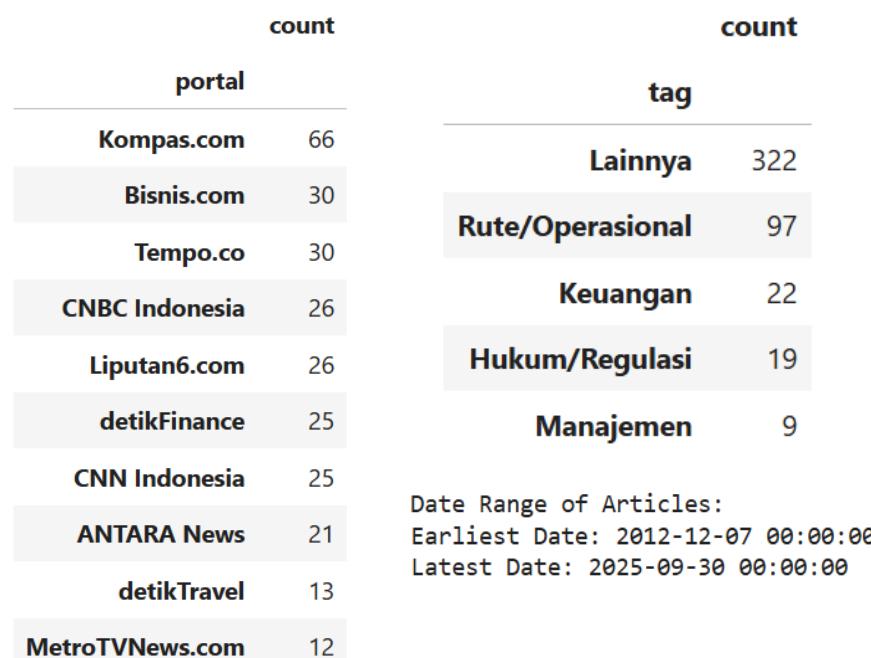


Gambar 4.14 Flag manual dengan Excel

Setelah pembersihan awal tersebut dilakukan, dilakukan pemeriksaan statistika deskriptif dasar terhadap data. Dari hasil pemeriksaan tersebut, didapatkan pengurangan jumlah baris data dari yang awalnya berjumlah 495 menjadi 469.

Descriptive Statistics:						
	link	judul	konten	tanggal	portal	tag
count	469	469	469	469	469	469
unique	469	469	469	176	90	5
top	https://www.tempo.co/ekonomi/tahun-depan-garud...	Tahun Depan Garuda Datangkan 24 Pesawat Baru	TEMPO.CO, Jakarta - Maskapai penerbangan pelat...	10/6/2025	Kompas.com	Lainnya
freq	1	1	1	19	66	322

Gambar 4.15 Distribusi hasil *scraping* konten setelah *cleaning manual*



Gambar 4.16 Distribusi hasil *scraping* konten setelah *cleaning manual* per portal dan tag

4.1.4 Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini tersusun atas beberapa kolom (atribut) yang merepresentasikan data-data tertentu yang akan digunakan dalam proses-proses selanjutnya. Definisi dari setiap kolom dataset dapat dilihat pada Tabel 5.1.

Tabel 4.2 Definisi dataset sebelum praproses

No.	Nama Kolom	Deskripsi
1	link	URL atau tautan unik yang merujuk langsung ke sumber artikel berita asli.

2	judul	Judul asli dari artikel berita yang diambil dari sumbernya.
3	konten	Isi atau teks lengkap dari artikel berita yang menjadi objek utama analisis.
4	tanggal	Tanggal publikasi artikel berita, menunjukkan kapan berita tersebut diterbitkan.
5	portal	Nama media atau portal berita yang mempublikasikan artikel.
6	tag	<p>Hasil <i>tagging</i> manual berdasarkan judul berita. Penentuan <i>tag</i> dilakukan berdasarkan kecocokan ekspresi reguler (<i>regex</i>) dari teks (setelah diubah menjadi huruf kecil) dengan daftar kata kunci yang telah ditentukan menggunakan fungsi =ARRAYFORMULA() di Google Spreadsheets. (Lihat bagian 4.1.2 di laporan)</p> <ul style="list-style-type: none"> • Tag "Keuangan" diberikan jika teks mengandung salah satu kata berikut: <i>laba, rugi, pendapatan, utang, restrukturisasi, pkpu, obligasi, kuartal</i>, atau <i>laporan keuangan</i>. • Tag "Rute/Operasional" diberikan jika teks mengandung salah satu kata berikut: <i>rute, penerbangan baru, buka rute, tutup rute, frekuensi penerbangan, operasional, insiden, delay, batal</i>, atau <i>keselamatan</i>. • Tag "Manajemen" diberikan jika teks mengandung salah satu kata berikut: <i>direktur, komisaris, manajemen, rups, ceo, pergantian, atau pengurus</i>. • Tag "Hukum/Regulasi" diberikan jika teks mengandung salah satu kata berikut: <i>kasus, pengadilan, kpk, dugaan, suap, hukum, sidang, sanksi, denda, atau izin</i>. • Tag "Lainnya" diberikan jika teks tidak memenuhi kriteria kata kunci dari salah satu kategori di atas.
7	sentiment	<p>Label hasil analisis polaritas sentimen kalimat: Positive, Neutral, atau Negative hasil klasifikasi GPT-4.1 berdasarkan konteks isi berita dengan verifikasi manual (Lihat subbab 4.2.2</p> <p> 2025-1_ProjectB_Klp-11_Paper).</p>

4.2 Hasil Analisis Sentimen Dasar

<<Soal No.1: Jelaskan bagaimana Anda akan mengklasifikasikan sentimen secara MANUAL dari setiap baris data pada dataset yang telah dikumpulkan sebelumnya. Sebutkan juga alasan linguistik atau konteks yang mendukung penentuan sentimen tersebut.>>

4.2.1 Tagging dan Klasifikasi Sentimen dengan OpenAI API

Untuk meningkatkan akurasi klasifikasi, dilakukan proses penandaan (*tagging*) dengan bantuan model bahasa besar (LLM) melalui OpenAI API. Model yang digunakan adalah GPT-4.1, dengan pertimbangan keseimbangan antara harga dan efektivitas pemrosesan teks dalam skala besar.

Selain menghasilkan kategori topik baru (`tag_new`), tahap ini juga menambahkan dimensi analisis sentimen, yaitu mengidentifikasi polaritas emosi dari teks (*positive*, *neutral*, atau *negative*). Dengan demikian, setiap berita akan memiliki dua informasi tambahan.

Proses analisis sentimen dilakukan menggunakan model bahasa GPT-4.1 yang diintegrasikan melalui API OpenAI pada Google Colab. Sebelum menjalankan analisis, dilakukan proses instalasi dependensi seperti openai, pandas, dan google-colab, serta inisialisasi variabel API agar koneksi dapat dijalankan secara otomatis. Setelah itu, dilakukan pengaturan nama file input dan output, penentuan kolom teks yang akan dianalisis, serta daftar kategori berita yang valid, yaitu *Kinerja & Keuangan*, *Operasional & Pelayanan*, *Regulasi & Kebijakan*, *Krisis & Kontroversi*, dan *Industri & Pariwisata Nasional*.

```
system_instruction = (
    "Kamu adalah AHLI analisis sentimen yang sangat AKURAT. "
    "Untuk setiap teks, kamu harus mengeluarkan output dalam format JSON lengkap seperti contoh berikut:\n\n"
    'Contoh:\n'
    '{"sentiment": "Positive", "tag_new": "Operasional & Pelayanan"}\n'
    '{"sentiment": "Neutral", "tag_new": "Kinerja & Keuangan"}\n'
    '{"sentiment": "Negative", "tag_new": "Krisis & Kontroversi"}\n\n'
    "Selalu isi kedua field dengan nilai yang sesuai.\n"
    "Nilai sentiment hanya boleh: Positive, Neutral, atau Negative.\n"
    "Nilai tag hanya boleh salah satu dari lima kategori berikut:\n"
    "1. Kinerja & Keuangan\n"
    "2. Operasional & Pelayanan\n"
    "3. Regulasi & Kebijakan\n"
    "4. Krisis & Kontroversi\n"
    "5. Industri & Pariwisata Nasional\n\n"
    "Jangan kosongkan field apapun. Jangan tulis apapun di luar JSON."
)
```

Gambar 4.17 Prompt labelling API OpenAI

Fungsi utama dalam tahap ini adalah `analyze_sentiment_openai()`. Fungsi ini menerima input berupa teks berita, kemudian mengirimkannya ke model GPT-4.1 untuk dianalisis. Model diminta menghasilkan output dalam format JSON yang berisi dua atribut, yaitu `sentiment` dan `tag_new`. Atribut `sentiment` hanya memiliki tiga kemungkinan nilai, yakni *Positive*, *Neutral*, atau *Negative*, sedangkan `tag_new` menentukan kategori berita sesuai konteks isi teks. Untuk memastikan hasil yang valid, fungsi ini juga mencakup proses validasi dan *fallback* agar sistem tetap memberikan hasil standar meskipun terjadi kesalahan pemrosesan.

```
--- UJI Coba Fungsi Sentimen ---
Menguji Teks 1...
Teks: 'Pelayanan di maskapai ini sangat memuaskan, saya a...'
Sentimen Hasil: **{'sentiment': 'Positive', 'tag_new': 'Operasional & Pelayanan'}**

-----
Menguji Teks 2...
Teks: 'Berita tentang pergantian direktur utama ini tidak...'
Sentimen Hasil: **{'sentiment': 'Neutral', 'tag_new': 'Kinerja & Keuangan'}**

-----
Menguji Teks 3...
Teks: 'Pembatalan penerbangan tanpa pemberitahuan adalah ...'
Sentimen Hasil: **{'sentiment': 'Negative', 'tag_new': 'Operasional & Pelayanan'}**

-----
Menguji Teks 4...
Teks: 'Seperti Labuan Bajo, Garuda mendukung pariwisata n...'
Sentimen Hasil: **{'sentiment': 'Negative', 'tag_new': 'Industri & Pariwisata Nasional'}**

---
--- Uji Coba Selesai ---
```

Gambar 4.18 Pengujian labelling awal

Sebelum dijalankan pada keseluruhan data, dilakukan uji coba fungsi analisis sentimen menggunakan empat contoh kalimat. Tujuan uji coba ini adalah memastikan bahwa fungsi dapat mengembalikan hasil analisis dalam format yang sesuai dan mampu membedakan jenis sentimen secara benar. Hasil uji coba ditampilkan pada gambar berikut yang menunjukkan keluaran di Google Colab saat fungsi dijalankan.

```
Memuat data dari: 5-garudaindonesia_news_stem.csv
Data dimuat. Jumlah baris: 469
Mulai menganalisis sentimen untuk 469 baris dengan model gpt-4.1.

Menganalisis Sentimen:  0% | 0/469 [00:00<?, ?it/s]

Analisis Sentimen dan Tagging Selesai! 🎉
```

Gambar 4.19 Tampilan akhir *labelling* dan *tagging*

```
Hasil Sentimen dan Tag (5 Baris Pertama):
    konten sentiment \
0 Garuda Indonesia Kembali RUPSLB di Tengah Isu ... Neutral
1 Garuda Gelar RUPSLB di Tengah Isu Masuknya Dir... Neutral
2 JAKARTA - Ketua Komisi V DPR Lasarus mengataka... Negative
3 Latar Belakang\nPada pertengahan 2023, wacana ... Neutral
4 --\nPlt Menteri Badan Usaha Milik Negara (BUMN...) Neutral

    tag_new
0 Kinerja & Keuangan
1 Regulasi & Kebijakan
2 Krisis & Kontroversi
3 Kinerja & Keuangan
4 Operasional & Pelayanan

-----
Data tersimpan ke file baru: 6-garudaindonesia_news_stem_tag_new_dan_sentiment.csv
```

Gambar 4.20 Preview data dengan *sentiment*

Setelah fungsi dipastikan berjalan dengan baik, tahap berikutnya adalah penerapan analisis sentimen terhadap keseluruhan dataset. Dataset dibaca dari file *5-garudaindonesia_news_stem.csv*, kemudian setiap baris teks dianalisis menggunakan fungsi yang sama dengan bantuan *progress bar* *tqdm* untuk memantau kemajuan proses. Hasil dari setiap baris disimpan ke dalam dua kolom baru, yaitu *sentiment* dan *tag_new*, lalu diekspor ke file keluaran *6-garudaindonesia_news_stem_tag_new_dan_sentiment.csv*.

```
⌚ Mengecek ulang baris yang gagal (API Error)...
⚠ Ditemukan 17 baris dengan API Error. Mengulang analisis...

⌚ Re-analyzing failed rows:  0% | 0/17 [00:00<?, ?it/s]

Re-analisis selesai. File '6-garudaindonesia_news_stem_tag_new_dan_sentiment.csv' telah diperbarui (termasuk kolom tag_new).
```

Gambar 4.21 Pengecekan sentiment dan tag

Langkah terakhir berupa proses pengecekan ulang terhadap kemungkinan baris yang gagal dianalisis akibat keterbatasan kuota atau *rate limit API*. Baris yang mengandung pesan kesalahan seperti *API Error* akan diproses kembali secara otomatis hingga seluruh data

berhasil dianalisis. Dengan demikian, hasil akhir yang tersimpan telah mencakup seluruh teks berita beserta label sentimen dan kategorinya dengan tingkat kelengkapan penuh.

Pada penggeraan Project B ini, kolom *tag_new* tidak digunakan. Hanya kolom *sentiment* yang digunakan untuk proses-proses selanjutnya, termasuk praproses data dan pelatihan model (Lihat deskripsi *dataset* di 4.1.4 2025-1_ProjectB_Klp-11_Paper).

4.2.2 Verifikasi Manual Penentuan Sentimen

Setelah proses penandaan sentimen dilakukan secara otomatis menggunakan OpenAI API, dilakukan tahap verifikasi manual terhadap hasil klasifikasi sentimen untuk memastikan kesesuaian label dengan konteks linguistik dan makna semantik dari setiap berita. Pada tahap ini, setiap baris data ditinjau kembali dengan membaca isi artikel secara menyeluruh dan mengevaluasi sentimen berdasarkan indikator linguistik seperti pemilihan kata bermada evaluatif (misalnya *meningkat, membaik, memburuk, kerugian, krisis*), struktur kalimat, serta konteks keseluruhan berita. Berita dikategorikan sebagai Positive apabila mengandung penilaian atau implikasi yang menguntungkan bagi Garuda Indonesia, Negative apabila menyoroti permasalahan, kerugian, atau kondisi krisis, dan Neutral apabila bersifat informatif atau faktual tanpa ekspresi penilaian emosional yang dominan. Hasil verifikasi menunjukkan bahwa label sentimen yang dihasilkan oleh model GPT-4.1 telah konsisten dengan interpretasi manual berdasarkan konteks dan alasan linguistik tersebut, sehingga tidak dilakukan perubahan label pada dataset akhir. Dengan demikian, proses ini memastikan bahwa data yang digunakan pada tahap pemodelan telah melalui kombinasi pendekatan otomatis dan validasi manusia (*human-in-the-loop*), yang meningkatkan reliabilitas penentuan sentimen.

4.3 Hasil Praproses Data

<<Soal No.2: Untuk analisis sentimen menggunakan model pretrained BERT, proses pre-processing yang dibutuhkan adalah berbeda dari pendekatan machine learning tradisional (SVM, Naive Bayes, dll.). BERT tidak membutuhkan pre-processing agresif karena ia dilatih dengan teks mentah. Uraikan apa langkah-langkah pra-pemrosesan teks yang paling tepat untuk dataset Anda jika akan digunakan sebagai data latih untuk model analisis sentimen berbasis model BERT. Jelaskan perbedaannya dengan model BERT vs pendekatan machine learning tradisional.>>

Tahap praproses data dilakukan untuk memastikan kualitas teks sebelum digunakan pada tahap pemodelan berbasis transformer. Data yang digunakan pada penelitian ini berupa artikel berita Garuda Indonesia yang diperoleh melalui proses web scraping dari berbagai sumber berita daring. Mengingat data hasil scraping berpotensi mengandung noise dan artefak teknis, praproses dilakukan secara bertahap dan selektif dengan tetap mempertahankan konteks semantik teks.

Praproses diawali dengan pembersihan data secara manual, yaitu penghapusan artikel yang tidak menggunakan Bahasa Indonesia, penghapusan baris kosong, serta penghapusan artikel dengan panjang teks yang tidak memadai. Tahap ini bertujuan untuk memastikan bahwa hanya artikel berita yang relevan dan representatif yang digunakan dalam proses

pelatihan model. Setelah itu, dilakukan normalisasi teks secara ringan (*light normalization*), mengingat karakteristik teks berita yang cenderung formal dan terstruktur.

4.3.1 Normalisasi *Slang*

Dilakukan normalisasi kosakata tidak baku menggunakan leksikon *colloquial Indonesian*, dengan menggantikan kata tidak baku atau bentuk slang yang terdeteksi ke dalam bentuk formalnya. Meskipun data berita umumnya bersifat formal, langkah ini tetap dilakukan sebagai upaya standarisasi teks dan untuk menjaga konsistensi antarartikel. Pada tahap ini, tanda baca dan simbol pemisah tetap dipertahankan karena memiliki peran penting dalam menjaga struktur kalimat dan membantu model berbasis *self-attention* memahami relasi antarkata.

```
lex = pd.read_csv("colloquial-indonesian-lexicon.csv")
lex = lex[['slang', 'formal']].dropna()
# normalisasi ringan
lex['slang'] = lex['slang'].str.lower().str.strip()
lex['formal'] = lex['formal'].str.lower().str.strip()

# kalau ada slang duplikat, ambil yang pertama
lex = lex.drop_duplicates(subset='slang')

# buat dictionary mapping
slang2formal = dict(zip(lex['slang'], lex['formal']))

import re

# urutkan slang berdasarkan panjang (desc)
sorted_slang = sorted(slang2formal.keys(), key=len, reverse=True)

# escape karakter khusus & gabungkan jadi regex
pattern = re.compile(
    r'\b(' + '|'.join(map(re.escape, sorted_slang)) + r')\b',
    flags=re.IGNORECASE
)

def normalize_slang(text):
    if not isinstance(text, str):
        return text

    def replace(match):
        return slang2formal.get(match.group(0).lower(), match.group(0))

    text = pattern.sub(replace, text)

    # rapihin spasi ganda
    text = re.sub(r'\s+', ' ', text).strip()
    return text
```

Gambar 4.22 Kode Normalisasi Slang

4.3.2 Pembersihan URL

Dilakukan penghapusan elemen non-konten yang berasal dari struktur situs berita, khususnya yang meliputi tautan URL yang muncul akibat proses scraping. Langkah ini penting agar model tidak memproses tautan sebagai bagian dari konten semantik berita.

```
import re

def remove_url(text):
    if not isinstance(text, str):
        return text

    # hapus http, https, www
    text = re.sub(r'https?://\S+|www\.\S+', '', text)
```

```
# rapihin spasi
text = re.sub(r'\s+', ' ', text).strip()
return text
```

Gambar 4.23 Kode Pembersihan URL

4.3.3 Pembersihan Call-to-action “Baca selanjutnya”

Fungsi `remove_news_cta` dirancang untuk membersihkan data teks dari elemen *boilerplate* atau *Call-to-Action* (CTA) yang umum ditemukan pada artikel berita daring, seperti frasa "baca selengkapnya", "baca juga", atau "klik di sini". Menggunakan *library Regular Expressions* (Regex), kode ini mengidentifikasi pola kata kunci navigasi tersebut dan menghapus seluruh segmen teks mulai dari kata kunci hingga akhir baris, karena informasi setelah frasa tersebut biasanya dianggap sebagai *noise* yang tidak relevan dengan konten utama. Selain penghapusan, fungsi ini juga melakukan validasi tipe data dan normalisasi *whitespace* untuk memastikan teks keluaran rapi dan bebas dari spasi ganda yang mungkin tertinggal pasca-pemotongan.

```
import re

def remove_news_cta(text):
    if not isinstance(text, str):
        return text

    cta_patterns = [
        r'\bbaca selengkapnya\b.*',
        r'\bbaca lebih lanjut\b.*',
        r'\bbaca juga\b.*',
        r'\bselengkapnya\b.*',
        r'\bklik di sini\b.*',
        r'\bklik disini\b.*'
    ]

    for pat in cta_patterns:
        text = re.sub(pat, ' ', text, flags=re.IGNORECASE)

    # rapihin spasi
    text = re.sub(r'\s+', ' ', text).strip()
    return text
```

Gambar 4.24 Kode Pembersihan Call to Action

4.3.4 Pembersihan *Encoding Artefacts*

Normalisasi teknis dilakukan dengan memperbaiki kesalahan *encoding* karakter yang muncul akibat proses scraping, seperti karakter kutip, simbol panah, dan karakter non-ASCII yang terdistorsi, menggunakan pustaka `fuzzy`. Untuk menangani sisa artefak *encoding* yang tidak sepenuhnya diperbaiki secara otomatis, dilakukan pembersihan tambahan terhadap karakter artefak yang umum muncul, seperti simbol hasil *double encoding*. Langkah ini bertujuan untuk memastikan bahwa teks yang digunakan bebas dari gangguan visual maupun token yang tidak bermakna secara linguistik.

```
pip install fuzzy
```

```

import ftfy

def fix_text_encoding(text):
    if not isinstance(text, str):
        return text
    return ftfy.fix_text(text)

import re

def remove_encoding_artifacts(text):
    if not isinstance(text, str):
        return text

    # 1) buang karakter "Ã" yang sering jadi sisa NBSP decode
    text = text.replace("Ã", " ")

    # 2) perbaiki mojibake kutip/dash yang umum (kalau masih ada)
    text = (text
            .replace("â€œ", '").replace("â€“, '",')
            .replace("â€˜", "'").replace("â€™", "'")
            .replace("â€“", "-").replace("â€”, “-”)
            .replace("â€¡", "..."))
    )

    # 3) buang bullet/arrow yang sering nyangkut
    text = re.sub(r"[•♦▶»◀]", " ", text)

    return text

```

Gambar 4.25 Kode Pembersihan Encoding

4.3.5 Pembersihan *Boilerplate*

Dilakukan penghapusan teks navigasi atau *boilerplate* seperti “Baca selengkapnya” dan “Baca juga” beserta variasinya yang muncul akibat proses scraping. Penghapusan elemen *boilerplate* ini bertujuan untuk menghilangkan informasi yang tidak memiliki kontribusi semantik terhadap isi berita dan dapat mengganggu proses tokenisasi.

```

def remove_news_boilerplate(text):
    if not isinstance(text, str):
        return text

    # tangkap "baca lebih lanjut/lajut", "baca selengkapnya", "baca juga", dll
    patterns = [
        r"\bbaca\s+lebih\s+lanj(?:ut|ut|u?t)?\b.*$",
        r"\bbaca\s+selengkapnya\b.*$",
        r"\bbaca\s+juga\b.*$",
        r"\bselengkapnya\b.*$",
        r"\bklik\s+di\s+ini\b.*$",
        r"\bklik\s+disini\b.*$",
    ]

    for p in patterns:
        text = re.sub(p, "", text, flags=re.IGNORECASE)

    return text

```

Gambar 4.26 Kode Pembersihan Boilerplate

4.3.6 Normalisasi *Whitespace*

Tahap praproses diakhiri dengan perapian spasi, yaitu penghapusan spasi ganda dan baris kosong yang muncul akibat proses scraping dan pembersihan sebelumnya. Seluruh data hasil praproses kemudian disimpan dalam format CSV sebagai *dataset* final.

```
def normalize_whitespace(text):
```

```

if not isinstance(text, str):
    return text
return re.sub(r'\s+', ' ', text).strip()
df["konten_clean_final"] = (
    df["konten"]
    .apply(fix_text_encoding)
    .apply(remove_encoding_artifacts)      # <-- pindah ke sini (lebih awal)
    .apply(normalize_slang)
    .apply(remove_url)
    .apply(remove_news_boilerplate)        # <-- sebelum whitespace final
    .apply(remove_news_cta)
    .apply(normalize_whitespace)          # <-- terakhir
)
mask      = df["konten_clean_final"].str.contains(r"(?:À|â€|â€œ|â€™|Â·|Â»)", regex=True,
na=False)
df.loc[mask, "konten_clean_final"].head(10)

print("Rows with artifacts:", mask.sum())
# buang baris kosong setelah cleaning
df = df.dropna(subset=['konten_clean_final'])
df = df[df['konten_clean_final'].str.strip() != '']

# (opsional) hapus duplikat konten
df = df.drop_duplicates(subset=['konten_clean_final'])

df.to_csv(
    "garuda_news_preprocessed_final.csv",
    index=False,
    encoding="utf-8"
)
df.head(5)

```

Gambar 4.27 Kode Normalisasi *Whitespace*

4.3.7 Pemetaan Label Kelas

Pada proses *fine-tuning* model IndoBERT untuk klasifikasi sentimen nantinya, diperlukan pemetaan antara label kategorikal (teks) dan label numerik (integer). Hal ini karena model *transformer* tidak memproses label dalam bentuk teks, melainkan dalam bentuk indeks numerik. Oleh karena itu, setiap kelas sentimen direpresentasikan menggunakan bilangan bulat yang konsisten selama proses pelatihan dan evaluasi.

Pada penelitian ini, digunakan tiga kelas sentimen, yaitu *Negative*, *Neutral*, dan *Positive*, yang dipetakan ke dalam bentuk numerik sebagai berikut:

- Negative → 0
 - Neutral → 1
 - Positive → 2

```

label2id = {"Negative": 0, "Neutral": 1, "Positive": 2}
id2label = {v: k for k, v in label2id.items()}

df["label"] = df[LABEL_COL].map(label2id)

# cek ada yang gagal mapping?
unmapped = df[df["label"].isna()][LABEL_COL].value_counts()
print("Unmapped labels:\n", unmapped)

df = df.dropna(subset=["label"]).copy()
df["label"] = df["label"].astype(int)

df["label"].value_counts()

```

Gambar 4.28 Kode Pemetaan Label Kelas

Pemetaan ini disimpan dalam dua struktur data, yaitu label2id dan id2label. Struktur label2id digunakan untuk mengonversi label teks menjadi label numerik yang dibutuhkan oleh model pada saat pelatihan. Sebaliknya, struktur id2label digunakan untuk mengonversi kembali hasil prediksi model dari bentuk numerik ke bentuk label teks yang lebih mudah diinterpretasikan pada tahap evaluasi dan analisis hasil.

Secara implementasi, pemetaan ini juga diberikan langsung ke model saat inisialisasi fine-tuning. Dengan cara ini, model dapat menghasilkan output prediksi yang konsisten dengan definisi kelas sentimen yang digunakan dalam penelitian, serta memudahkan proses pelaporan hasil seperti perhitungan metrik evaluasi dan analisis kesalahan prediksi.

4.3.8 Pembagian Dataset

Sebelum proses tokenisasi dan pelatihan model dilakukan, dataset terlebih dahulu dibagi ke dalam tiga subset, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*test set*). Pembagian ini bertujuan untuk memastikan bahwa proses pelatihan, pemilihan model, dan evaluasi akhir dilakukan secara terpisah sehingga dapat mencerminkan kemampuan generalisasi model secara objektif.

```

from sklearn.model_selection import train_test_split

train_df, temp_df = train_test_split(
    df,
    test_size=0.2,
    random_state=42,
    stratify=df["label"]
)

val_df, test_df = train_test_split(
    temp_df,
    test_size=0.5,
    random_state=42,
    stratify=temp_df["label"]
)

```

Gambar 4.29 Kode Pembagian Dataset

Pada penelitian ini, pembagian data dilakukan secara bertahap menggunakan fungsi `train_test_split` dari pustaka *scikit-learn*. Tahap pertama membagi *dataset* menjadi 80% data latih dan 20% data sementara. Selanjutnya, data sementara tersebut dibagi kembali menjadi 10% data validasi dan 10% data uji, sehingga menghasilkan proporsi akhir sebesar 80%:10%:10% untuk data latih, validasi, dan uji.

Tabel 4.3 Pembagian Dataset

Pembagian Dataset	Proporsi	Pembagian Dataset	Proporsi
Tahap 1		Tahap 2	
Data latih	80%	Data latih	80%
Data sementara	20%	Data validasi	10%
		Data uji	10%
Proporsi Akhir: 80:10:10			

Untuk menjaga keseimbangan distribusi kelas sentimen pada setiap subset, proses pembagian data dilakukan menggunakan teknik *stratified splitting*, dengan parameter `stratify` berdasarkan label sentimen. Pendekatan ini memastikan bahwa proporsi kelas negatif, netral, dan positif pada data latih, validasi, dan uji tetap mendekati distribusi kelas pada dataset awal. Hal ini penting untuk mencegah bias pelatihan dan memastikan bahwa evaluasi model tidak dipengaruhi oleh ketidakseimbangan kelas pada subset tertentu.

Selain itu, parameter `random_state` digunakan untuk memastikan bahwa proses pembagian data bersifat *reproducible*, sehingga eksperimen dapat diulang dengan hasil pembagian data yang konsisten. Setelah pembagian data dilakukan, distribusi label pada masing-masing subset diverifikasi dengan menghitung proporsi kelas untuk memastikan bahwa stratifikasi berjalan dengan baik.

Dengan pendekatan ini, proses pelatihan model dilakukan menggunakan data latih, pemilihan dan penyetelan *hyperparameter* dilakukan berdasarkan performa pada data validasi, dan evaluasi akhir model dilakukan pada data uji yang sepenuhnya terpisah dari proses pelatihan.

4.3.9 Tokenisasi Teks

Setelah dataset dibagi menjadi data latih, validasi, dan uji, tahap selanjutnya adalah tokenisasi teks untuk mengubah teks mentah menjadi representasi numerik yang dapat diproses oleh model IndoBERT. Pada penelitian ini digunakan *tokenizer* bawaan IndoBERT yang diakses melalui kelas AutoTokenizer. Tokenizer ini telah dilatih menggunakan skema tokenisasi berbasis *subword*, sehingga mampu menangani variasi kosakata dan morfologi Bahasa Indonesia secara efektif.

Tokenisasi dilakukan dengan cara mengonversi setiap teks artikel berita menjadi urutan token yang kemudian direpresentasikan dalam bentuk indeks numerik (input_ids). Proses ini juga menghasilkan attention_mask yang menandai bagian token yang relevan untuk diproses oleh model. Parameter truncation=True digunakan untuk memastikan bahwa teks yang melebihi panjang maksimum input akan dipotong, sehingga tidak melampaui batas kapasitas model. Panjang maksimum input (max_length) ditetapkan sesuai dengan konfigurasi eksperimen yang digunakan.

```
from transformers import AutoTokenizer

def tokenize_texts(texts):
    return tokenizer(
        texts.tolist(),
        truncation=True,
        padding=False,      # padding nanti pak
        max_length=MAX_LEN
    )

train_enc = tokenize_texts(train_df[TEXT_COL])
val_enc   = tokenize_texts(val_df[TEXT_COL])
test_enc  = tokenize_texts(test_df[TEXT_COL])
```

Gambar 4.29 Kode Pembagian Dataset

Pada tahap tokenisasi ini, *padding* tidak dilakukan secara langsung (padding=False). Sebagai gantinya, padding diterapkan secara dinamis pada saat pelatihan menggunakan *DataCollator*. Pendekatan ini dipilih karena lebih efisien secara komputasi, mengingat panjang teks pada setiap batch dapat bervariasi dan padding hanya diterapkan hingga panjang maksimum pada batch tersebut, bukan pada seluruh *dataset*.

Tokenisasi dilakukan secara terpisah untuk data latih, validasi, dan uji, sehingga setiap subset data memiliki representasi token yang konsisten dengan pembagian dataset sebelumnya. Dengan demikian, proses tokenisasi ini memastikan bahwa seluruh teks telah dikonversi ke dalam format input yang sesuai dengan arsitektur IndoBERT sebelum digunakan pada tahap pelatihan dan evaluasi model.

4.3.10 Pembentukan *Dataset PyTorch*

Setelah proses tokenisasi selesai, data perlu dikonversi ke dalam format yang kompatibel dengan *framework* PyTorch agar dapat digunakan dalam proses pelatihan model. Pada penelitian ini, dibentuk sebuah kelas dataset kustom bernama NewsDataset yang mewarisi kelas Dataset dari pustaka torch.utils.data.

```

class NewsDataset(Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels.tolist()

    def __len__(self):
        return len(self.labels)

    def __getitem__(self, idx):
        item = {k: torch.tensor(v[idx]) for k, v in self.encodings.items()}
        item["labels"] = torch.tensor(self.labels[idx], dtype=torch.long)
        return item

train_dataset = NewsDataset(train_enc, train_df["label"])
val_dataset = NewsDataset(val_enc, val_df["label"])
test_dataset = NewsDataset(test_enc, test_df["label"])

```

Gambar 4.30 Kode Pembagian Dataset

Kelas NewsDataset bertugas untuk menggabungkan hasil tokenisasi teks dengan label sentimen yang sesuai pada setiap sampel. Hasil tokenisasi disimpan dalam bentuk *dictionary* yang berisi komponen input seperti `input_ids` dan `attention_mask`, sedangkan label sentimen disimpan dalam bentuk numerik. Pada metode `__getitem__`, setiap elemen input dikonversi menjadi tensor PyTorch, dan label dikonversi ke tipe `torch.long` agar sesuai dengan kebutuhan fungsi `loss` yang digunakan dalam klasifikasi multikelas.

Metode `__len__` digunakan untuk mengembalikan jumlah total sampel dalam dataset, sedangkan metode `__getitem__` memungkinkan pengambilan satu sampel data berdasarkan indeks tertentu. Dengan implementasi ini, dataset dapat diakses secara *lazy loading* oleh *DataLoader* atau *Trainer* tanpa harus memuat seluruh data ke memori sekaligus.

Dataset kustom ini kemudian diinisialisasi secara terpisah untuk data latih, validasi, dan uji, menggunakan hasil tokenisasi dan label dari masing-masing subset data. Pendekatan ini memastikan bahwa setiap subset data memiliki representasi input dan label yang konsisten, serta dapat digunakan secara langsung pada proses fine-tuning dan evaluasi model IndoBERT menggunakan API *Trainer* dari pustaka Transformers.

4.4. Hasil Pemilihan Model

<<Soal No.3: Model BERT apa yang paling tepat digunakan melakukan analisis sentiment untuk dataset Anda? Sertakan alasan terkait ukuran model, domain bahasa, dan trade-off performa terhadap kecepatan.>>

Model yang paling tepat digunakan dalam penelitian ini adalah IndoBERT-base-p2. Dari aspek domain bahasa, model ini dipilih karena dilatih secara khusus pada korpus Bahasa Indonesia, sehingga mampu menangkap representasi semantik dan nuansa linguistik kontekstual yang lebih baik dibandingkan model multilingual seperti mBERT. Selain itu, karena *dataset* penelitian berupa artikel berita formal dengan struktur kalimat panjang, IndoBERT dinilai lebih sesuai dibandingkan IndoBERTweet yang berbasis media sosial, sehingga potensi penurunan performa akibat *domain mismatch* dapat dihindari.

Pemilihan varian ini juga mempertimbangkan aspek ukuran model. Meskipun model transformer lain dengan ukuran yang lebih besar sempat dipertimbangkan, model-model

tersebut tidak digunakan karena adanya keterbatasan sumber daya komputasi serta risiko *overfitting* pada dataset yang ukurannya terbatas. Oleh karena itu, varian *base* dianggap cukup memadai untuk menangani kompleksitas data tanpa membebani sumber daya yang ada.

Terakhir, terkait *trade-off* performa terhadap kecepatan, IndoBERT-base-p2 menawarkan keseimbangan yang optimal. Versi p2 ini merupakan hasil *pretraining* lanjutan dari versi awal (p1) yang menghasilkan representasi bahasa yang lebih stabil dan matang, sehingga performanya lebih konsisten pada berbagai tugas *downstream*. Dengan mempertimbangkan efisiensi pelatihan dan konsistensi hasil, model ini menjadi pilihan yang paling seimbang dan tepat untuk diimplementasikan.

4.5 Hasil Fine-Tuning Model

<<Soal No.4: Jelaskan langkah-langkah utama melakukan fine-tuning BERT untuk kelas sentimen dataset Anda (misal: positif, negatif, netral) menggunakan HuggingFace Transformers. Minimal jelaskan: Tokenisasi, Loss function yang digunakan, Evaluasi model.>>

Bagian ini memaparkan implementasi dan hasil *fine-tuning* model BERT yang dilakukan menggunakan kerangka kerja HuggingFace Transformers. Proses adaptasi model terhadap *dataset* sentimen (positif, negatif, dan netral) dilakukan melalui serangkaian langkah sistematis, dimulai dari tokenisasi input agar sesuai dengan arsitektur model, penerapan fungsi *loss* yang relevan untuk mengoptimalkan bobot model selama pelatihan, hingga tahap evaluasi untuk mengukur performa model pada data uji. Berikut adalah rincian eksperimen yang diawali dengan pemodelan *baseline*.

4.5.1 Fungsi *Loss* yang Digunakan

Pada penelitian ini, proses fine-tuning model IndoBERT untuk tugas klasifikasi sentimen menggunakan Cross-Entropy Loss sebagai fungsi objektif. Fungsi loss ini secara default diterapkan oleh arsitektur AutoModelForSequenceClassification pada pustaka *Transformers* ketika label kelas disertakan dalam proses pelatihan. Cross-Entropy Loss umum digunakan pada permasalahan klasifikasi *multiclass* karena mampu mengukur perbedaan antara distribusi probabilitas prediksi model dan distribusi label sebenarnya secara efektif.

Secara matematis, model menghasilkan skor mentah (*logits*) untuk setiap kelas sentimen K pada sebuah input teks, yang dinyatakan sebagai vektor:

$$\mathbf{z} = [z_1, z_2, \dots, z_K],$$

Gambar 4.31 Persamaan Vektor

dengan K adalah jumlah kelas. Skor tersebut kemudian diubah menjadi probabilitas menggunakan fungsi softmax sebagai berikut:

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}$$

Gambar 4.32 Persamaan Softmax

Nilai Cross-Entropy Loss untuk satu sampel dengan label sebenarnya y dihitung sebagai negatif log-probabilitas dari kelas yang benar:

$$\mathcal{L} = -\log(p_y)$$

Gambar 4.33 Persamaan Entropy Loss

Dalam konteks klasifikasi sentimen tiga kelas (negatif, netral, dan positif), fungsi loss ini memberikan penalti yang lebih besar ketika model memberikan probabilitas rendah pada kelas yang seharusnya, terutama apabila prediksi yang salah disertai dengan tingkat keyakinan yang tinggi. Sebaliknya, jika model mampu memprediksi kelas yang benar dengan probabilitas tinggi, nilai loss akan menjadi kecil.

Selama proses pelatihan, nilai Cross-Entropy Loss dihitung pada setiap batch data dan digunakan sebagai sinyal kesalahan untuk melakukan pembaruan bobot model melalui mekanisme *backpropagation*. Tujuan optimisasi dari proses *fine-tuning* ini adalah meminimalkan nilai loss secara keseluruhan sehingga distribusi probabilitas prediksi model semakin mendekati distribusi label sentimen yang sebenarnya. Penggunaan Cross-Entropy Loss pada penelitian ini sejalan dengan praktik umum dalam fine-tuning model *transformer* untuk klasifikasi teks dan memberikan dasar yang kuat bagi proses pembelajaran yang stabil dan efektif.

4.5.2 Pemodelan *Baseline*

Eksperimen awal pada penelitian ini bertujuan untuk memperoleh baseline performa model sebelum dilakukan penyesuaian atau eksperimen lanjutan. Pada tahap ini digunakan pendekatan fine-tuning standar (level 0) dengan model IndoBERT-base-p2, tanpa penerapan strategi lanjutan seperti *discriminative learning rate* atau *layer freezing*. Seluruh parameter model dilatih ulang (*full fine-tuning*) agar model dapat beradaptasi secara optimal terhadap tugas analisis sentimen pada domain berita.

Dataset yang telah melalui tahap praproses dibagi menjadi data latih, validasi, dan uji dengan rasio 80:10:10. Proses fine-tuning dilakukan menggunakan fungsi loss *Cross-Entropy* untuk klasifikasi multi-kelas tiga label, yaitu negatif, netral, dan positif. Parameter pelatihan yang digunakan meliputi *learning rate* sebesar 2e-5, ukuran *batch* pelatihan sebesar 16, ukuran *batch* evaluasi sebesar 32, serta regularisasi *weight decay* sebesar 0.01. Model dilatih selama maksimal 10 epoch dengan evaluasi pada setiap epoch menggunakan data validasi.

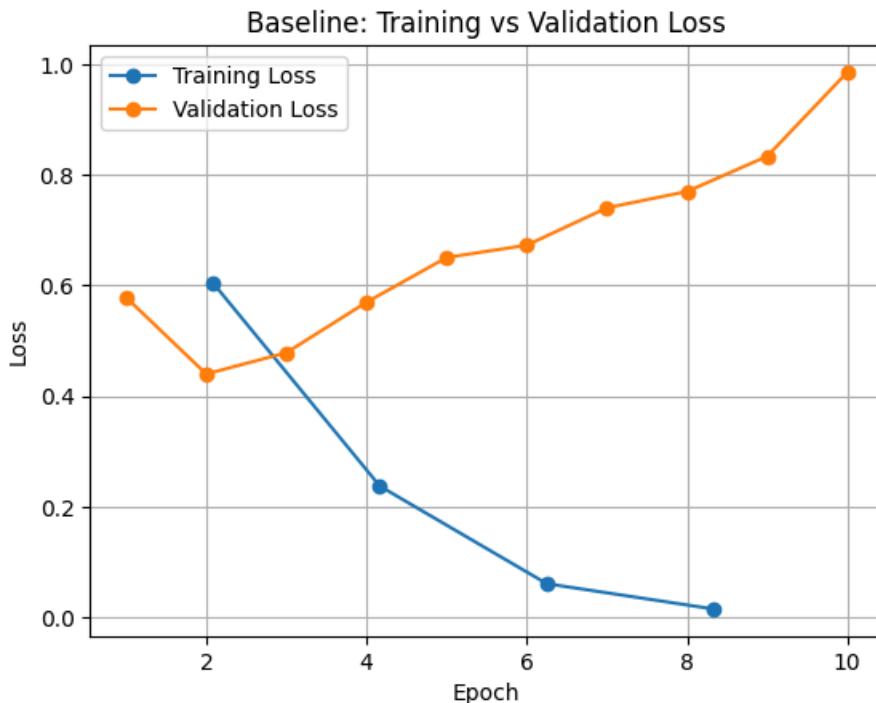
Evaluasi performa model dilakukan dengan menggunakan metrik *accuracy*, *macro-precision*, *macro-recall*, dan *macro-F1*. Pemilihan metrik *macro* dilakukan untuk memberikan penilaian yang seimbang terhadap seluruh kelas sentimen, termasuk kelas netral yang memiliki jumlah sampel lebih sedikit.

Tabel 4.4 Hasil Eksperimen Baseline

Epoch	Training Loss	Validation Loss	Accuracy	Precision Macro	Recall Macro	F1 Macro
1	No log	0.578438	0.744681	0.497558	0.612963	0.548867
2	No log	0.439855	0.808511	0.754545	0.724074	0.725146
3	0.604800	0.478337	0.808511	0.768519	0.762963	0.764133
4	0.604800	0.569706	0.744681	0.645733	0.650000	0.642722
5	0.237000	0.650600	0.829787	0.795587	0.762963	0.768803
6	0.237000	0.672870	0.829787	0.795587	0.762963	0.768803
7	0.060500	0.740504	0.851064	0.829004	0.800000	0.807456
8	0.060500	0.770037	0.851064	0.829004	0.800000	0.807456
9	0.014700	0.833749	0.851064	0.829004	0.800000	0.807456
10	0.014700	0.823622	0.851064	0.829004	0.800000	0.807456

Berdasarkan hasil *fine-tuning baseline* yang ditunjukkan pada tabel pelatihan, terlihat bahwa model mengalami penurunan *training loss* yang signifikan seiring bertambahnya epoch. Mulai dari epoch ke-3 hingga epoch ke-10, *training loss* menurun secara konsisten dari sekitar 0.60 hingga mendekati nol (0.0147). Penurunan ini menunjukkan bahwa model mampu mempelajari pola sentimen pada data latih dengan baik dan menyesuaikan bobotnya secara efektif selama proses pelatihan.

Sebaliknya, *validation loss* menunjukkan tren peningkatan setelah beberapa epoch awal. Nilai *validation loss* berada pada kisaran 0.44–0.57 pada epoch awal dan meningkat hingga di atas 0.82 pada epoch-epoch akhir. Pola ini mengindikasikan munculnya overfitting ringan, di mana model menjadi semakin spesifik terhadap data latih, namun tidak disertai dengan penurunan kinerja klasifikasi yang signifikan pada data validasi. Fenomena ini umum terjadi pada proses fine-tuning model berbasis *transformer* dengan ukuran dataset yang relatif terbatas.



Gambar 4.34 Perbandingan Training dan Validation Loss Baseline

Meskipun *validation loss* meningkat, metrik evaluasi utama, khususnya macro F1-score, menunjukkan tren peningkatan yang jelas. Nilai *macro F1* meningkat dari sekitar 0.55 pada epoch awal dan mencapai nilai tertinggi sebesar 0.807 pada epoch ke-7, kemudian bertahan stabil hingga epoch ke-10. Stabilitas nilai *macro F1* ini menunjukkan bahwa model telah mencapai titik saturasi pembelajaran, di mana penambahan epoch tidak lagi memberikan peningkatan kinerja klasifikasi yang signifikan.

Penurunan *training loss* yang berkelanjutan bersamaan dengan stabilnya nilai *macro F1* mengindikasikan bahwa meskipun model semakin yakin terhadap prediksinya pada data latih, kemampuan generalisasi terhadap data validasi tetap terjaga. Peningkatan *validation loss* pada kondisi ini lebih merefleksikan meningkatnya tingkat kepercayaan (*confidence*) model pada sejumlah kecil prediksi yang salah, terutama pada kelas dengan jumlah sampel terbatas, dibandingkan penurunan kemampuan klasifikasi secara keseluruhan.

Berdasarkan analisis tersebut, epoch ke-7 dapat dianggap sebagai titik pelatihan yang optimal pada eksperimen baseline, karena menghasilkan nilai *macro F1* tertinggi dengan keseimbangan yang baik antara kemampuan pembelajaran dan generalisasi. Hasil eksperimen baseline ini selanjutnya digunakan sebagai acuan utama dalam mengevaluasi efektivitas eksperimen lanjutan.

Berdasarkan hasil eksperimen *baseline*, model IndoBERT-base-p2 menunjukkan kinerja klasifikasi yang stabil dengan nilai *macro F1* yang mencapai titik saturasi pada epoch menengah. Meskipun *training loss* terus menurun, peningkatan *validation loss* mengindikasikan bahwa penambahan epoch selanjutnya tidak lagi memberikan peningkatan performa yang signifikan dan berpotensi meningkatkan tingkat overfitting. Kondisi ini menunjukkan bahwa keterbatasan kinerja model pada eksperimen baseline bukan disebabkan

oleh durasi pelatihan, melainkan oleh keterbatasan representasi input dan konfigurasi pelatihan yang digunakan.

Oleh karena itu, eksperimen lanjutan difokuskan pada penyesuaian aspek-aspek yang secara langsung memengaruhi kualitas representasi teks dan stabilitas pelatihan, bukan sekadar penambahan jumlah epoch. Penyesuaian tersebut meliputi pengaturan panjang input untuk mengurangi dampak pemotongan teks (*truncation*), penerapan strategi stabilisasi pelatihan melalui *learning rate scheduling* dan *warmup*, serta eksplorasi nilai *learning rate* yang lebih sesuai dengan ukuran dataset dan karakteristik teks berita. Melalui pendekatan ini, diharapkan model dapat memanfaatkan konteks yang lebih lengkap dan meningkatkan kemampuan generalisasi tanpa meningkatkan risiko overfitting secara berlebihan.

4.5.3 Hasil Evaluasi Model

Pada subbab ini, dipaparkan hasil kinerja model klasifikasi sentimen berdasarkan skenario eksperimen yang telah dirancang sebelumnya. Evaluasi dilakukan secara komparatif untuk menganalisis dampak dari setiap penyesuaian parameter terhadap efektivitas model dalam memprediksi kelas sentimen yang tepat. Seluruh hasil pengujian diukur menggunakan metrik evaluasi standar, meliputi *accuracy*, *precision*, *recall*, dan *F1-score*, guna memberikan gambaran performa yang menyeluruh. Pembahasan hasil evaluasi diawali dengan eksperimen mengenai pengaruh batasan panjang token input terhadap kemampuan model dalam menangkap konteks berita.

4.5.3.1 Eksperimen 1: Penyesuaian Panjang Input Teks

Eksperimen pertama bertujuan untuk mengevaluasi pengaruh panjang input teks (*max_length*) terhadap kinerja model IndoBERT dalam melakukan klasifikasi sentimen pada artikel berita. Pada eksperimen baseline, panjang input dibatasi pada jumlah token tertentu sehingga artikel berita yang memiliki panjang lebih besar berpotensi mengalami pemotongan (*truncation*). Pemotongan ini dapat menyebabkan hilangnya konteks penting yang diperlukan untuk memahami sentimen secara utuh, terutama pada berita yang mengandung opini implisit atau membutuhkan pemahaman lintas paragraf.

Pada eksperimen ini, batas panjang token ditingkatkan secara bertahap untuk memungkinkan model menerima konteks yang lebih lengkap dari setiap artikel. Dengan meningkatkan *max_length*, diharapkan representasi teks yang dihasilkan oleh model menjadi lebih informatif dan mampu menangkap hubungan semantik antar kalimat dengan lebih baik. Penyesuaian ini dilakukan tanpa mengubah komponen pelatihan lainnya, seperti arsitektur model, pembagian data, dan konfigurasi pelatihan, sehingga dampak perubahan panjang input dapat diamati secara terisolasi.

Evaluasi kinerja pada eksperimen ini tetap menggunakan metrik yang sama dengan eksperimen *baseline*, yaitu *accuracy*, *macro-precision*, *macro-recall*, dan *macro F1-score*. Fokus utama evaluasi adalah perubahan pada *macro F1*, mengingat metrik ini memberikan gambaran yang lebih seimbang terhadap performa model pada seluruh kelas sentimen, termasuk kelas dengan jumlah data yang lebih sedikit. Hasil dari eksperimen ini selanjutnya dibandingkan dengan baseline untuk menentukan apakah peningkatan panjang input mampu memberikan perbaikan kinerja yang signifikan.

Tabel 4.5 Hasil Eksperimen Penyesuaian Panjang Teks

Epoch	Training Loss	Validation Loss	Accuracy	Precision Macro	Recall Macro	F1 Macro
1	No log	0.568193	0.787234	0.525362	0.650000	0.580288
2	No log	0.473881	0.872340	0.907648	0.798148	0.814071
3	0.616200	0.512338	0.851064	0.832298	0.801852	0.807323
4	0.616200	0.528236	0.829787	0.787698	0.779630	0.782799
5	0.229100	0.676327	0.787234	0.724880	0.707407	0.707692
6	0.229100	0.656586	0.829787	0.795517	0.801852	0.797060
7	0.060100	0.756417	0.851064	0.822222	0.818519	0.818713
8	0.060100	0.772250	0.829787	0.800000	0.801852	0.797534
9	0.015500	0.761160	0.829787	0.790789	0.781481	0.784472
10	0.015500	0.776306	0.829787	0.790789	0.781481	0.784472

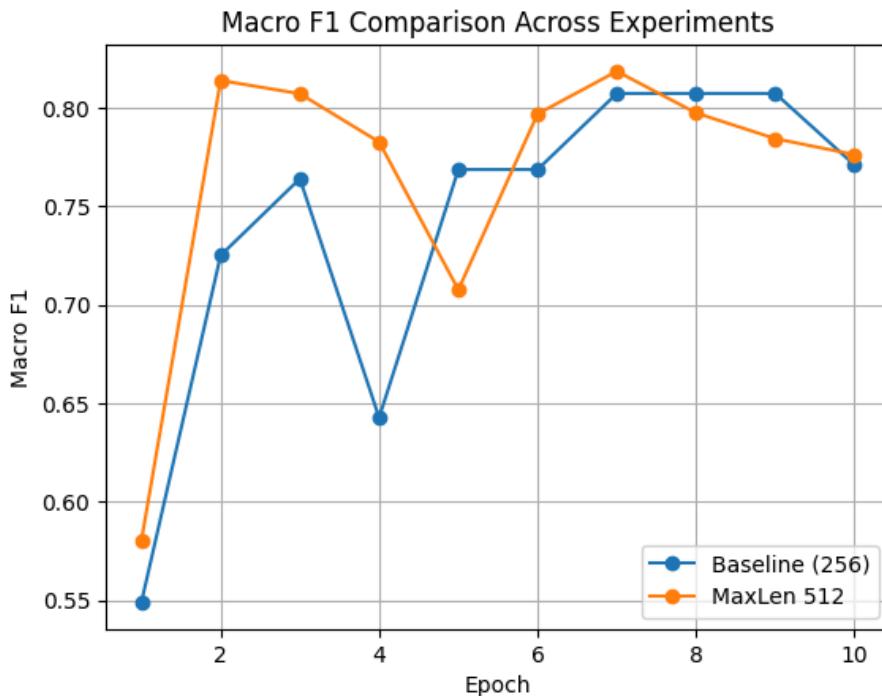
Eksperimen pertama mengevaluasi pengaruh peningkatan panjang input teks terhadap kinerja model IndoBERT. Dengan menetapkan `max_length` sebesar 512 token, model diberikan konteks teks yang lebih panjang dibandingkan eksperimen baseline, sehingga diharapkan mampu menangkap informasi sentimen yang tersebar di seluruh artikel berita.

Berdasarkan hasil pelatihan, terlihat bahwa nilai *macro F1* mengalami peningkatan signifikan pada epoch awal. Pada epoch ke-2, *macro F1* mencapai nilai 0.814, yang lebih tinggi dibandingkan nilai tertinggi pada eksperimen baseline. Hal ini menunjukkan bahwa peningkatan panjang input memungkinkan model memanfaatkan konteks tambahan untuk memperbaiki prediksi sentimen, khususnya pada artikel yang membutuhkan pemahaman lintas kalimat.



Gambar 4.35 Perbandingan Cross Entropy Loss Baseline dan MaxLen512

Namun, setelah epoch awal, nilai *macro F1* menunjukkan fluktuasi dan cenderung menurun pada epoch selanjutnya sebelum kembali meningkat secara terbatas. Pola ini mengindikasikan bahwa meskipun konteks tambahan membantu model pada tahap awal pelatihan, peningkatan panjang input juga meningkatkan kompleksitas representasi dan risiko *overfitting*, terutama pada dataset dengan ukuran terbatas. Hal ini tercermin dari *training loss* yang relatif tinggi pada epoch awal dan *validation loss* yang meningkat pada epoch-epoch selanjutnya.



Gambar 4.36 Perbandingan Macro F1 Baseline dan MaxLen512

Secara keseluruhan, performa terbaik pada eksperimen ini dicapai pada epoch ke-2 hingga ke-3, dengan nilai *macro F1* berkisar antara 0.807 hingga 0.814. Meskipun nilai tersebut lebih tinggi dibandingkan baseline, performa model tidak menunjukkan kestabilan yang konsisten pada epoch-epoch selanjutnya. Hal ini menunjukkan bahwa peningkatan `max_length` menjadi 512 token memberikan manfaat tambahan dalam menangkap konteks, namun juga memerlukan strategi pelatihan tambahan, seperti *early stopping* atau stabilisasi pelatihan, agar peningkatan performa dapat dipertahankan.

Dibandingkan dengan eksperimen baseline, eksperimen ini menunjukkan bahwa panjang input merupakan faktor penting dalam klasifikasi sentimen artikel berita. Namun, pemilihan `max_length` yang terlalu besar tanpa penyesuaian strategi pelatihan dapat menyebabkan fluktuasi performa, sehingga perlu dievaluasi lebih lanjut pada eksperimen lanjutan.

4.5.3.2 Eksperimen 2: Stabilisasi *Fine-Tuning* dengan *Warmup* dan *Learning Rate Scheduler*

Eksperimen kedua bertujuan untuk meningkatkan stabilitas proses fine-tuning model IndoBERT pada dataset berita dengan menerapkan warmup dan learning rate scheduler. Pada proses *fine-tuning* transformer, fase awal pelatihan sering bersifat sensitif karena model bertransisi dari representasi hasil pretraining menuju penyesuaian terhadap tugas klasifikasi yang spesifik. Jika pembaruan bobot terlalu agresif di awal, pelatihan dapat menjadi tidak stabil dan berpotensi menurunkan kemampuan generalisasi, terutama ketika ukuran dataset relatif terbatas.

Untuk mengatasi hal tersebut, eksperimen ini menambahkan mekanisme warmup, yaitu periode awal pelatihan di mana learning rate dinaikkan secara bertahap dari nilai kecil

hingga mencapai nilai target. Strategi ini membantu mencegah perubahan parameter yang terlalu besar pada langkah-langkah awal, sehingga model dapat beradaptasi secara lebih halus. Setelah fase warmup, digunakan learning rate scheduler (misalnya linear decay) untuk menurunkan learning rate secara bertahap seiring bertambahnya langkah pelatihan. Penurunan learning rate ini bertujuan meningkatkan konvergensi dan mengurangi risiko overfitting pada epoch-epoch akhir.

Pada eksperimen ini, komponen lain seperti model yang digunakan, pembagian dataset train/validasi/uji, serta metrik evaluasi tetap dipertahankan sama dengan baseline agar pengaruh warmup dan scheduler dapat diamati secara terisolasi. Evaluasi dilakukan menggunakan metrik accuracy serta macro-precision, macro-recall, dan macro F1-score, dengan fokus utama pada *macro F1* sebagai indikator performa yang seimbang untuk seluruh kelas sentimen.

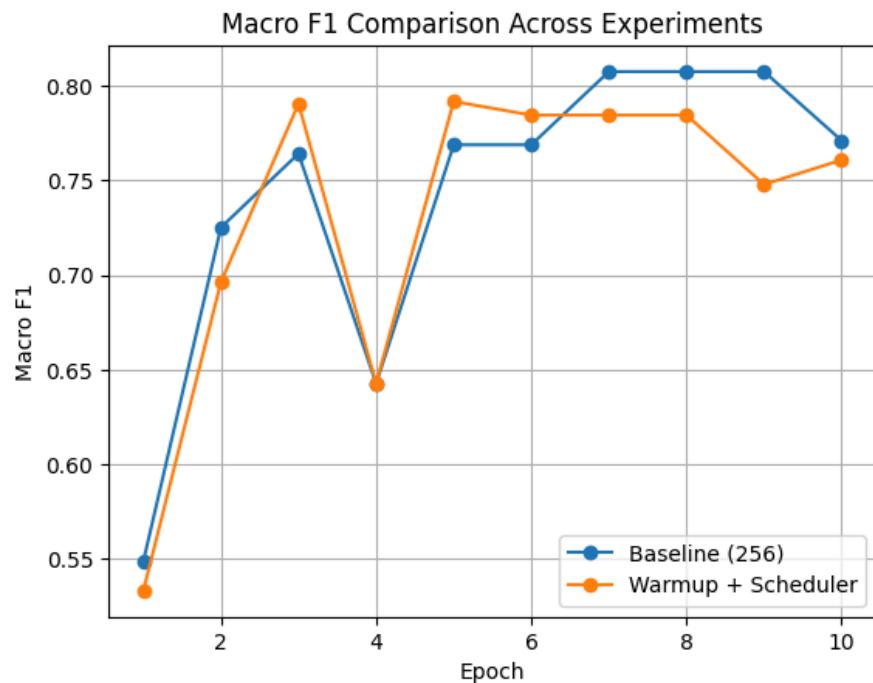
Tabel 4.6 Hasil Eksperimen *Warmup* dan *Learning Rate Scheduler*

Epoch	Training Loss	Validation Loss	Accuracy	Precision Macro	Recall Macro	F1 Macro
1	No log	0.879441	0.723404	0.482424	0.596296	0.533333
2	No log	0.524529	0.808511	0.767426	0.705556	0.696283
3	0.819400	0.475789	0.851064	0.840909	0.779630	0.790476
4	0.819400	0.506204	0.744681	0.645733	0.650000	0.642722
5	0.332600	0.618463	0.851064	0.846377	0.779630	0.791685
6	0.332600	0.533154	0.829787	0.790789	0.781481	0.784472
7	0.106100	0.607821	0.829787	0.790789	0.781481	0.784472
8	0.106100	0.663767	0.829787	0.790789	0.781481	0.784472
9	0.023900	0.712140	0.808511	0.760234	0.744444	0.747863
10	0.023900	0.719451	0.808511	0.760234	0.744444	0.747863

Eksperimen kedua mengevaluasi pengaruh penerapan warmup dan learning rate scheduler linear terhadap stabilitas dan kinerja fine-tuning model IndoBERT pada konfigurasi panjang input standar (`max_length = 256`). Tujuan utama eksperimen ini adalah untuk mengurangi ketidakstabilan pembaruan bobot pada fase awal pelatihan dan meningkatkan kemampuan generalisasi model.

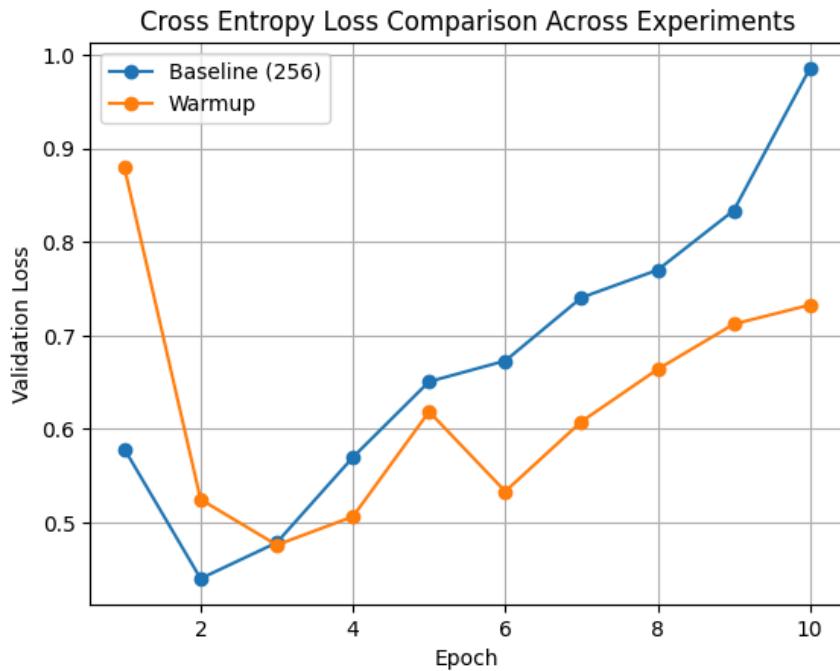
Berdasarkan hasil pelatihan, terlihat bahwa nilai *training loss* pada epoch awal relatif tinggi dan kemudian menurun secara bertahap, yang menunjukkan bahwa proses pembelajaran berlangsung lebih konservatif dibandingkan eksperimen baseline. Hal ini

merupakan karakteristik yang diharapkan dari penerapan warmup, di mana learning rate dinaikkan secara bertahap sehingga model tidak langsung melakukan pembaruan parameter secara agresif pada langkah awal.



Gambar 4.37 Perbandingan Macro F1 Baseline dan Warmup+Scheduler

Nilai *macro F1* menunjukkan peningkatan yang cukup signifikan pada epoch ke-3, mencapai nilai sekitar 0.790, yang sebanding dengan performa terbaik pada eksperimen baseline. Namun, setelah mencapai nilai tersebut, performa *macro F1* mengalami fluktuasi pada epoch berikutnya, dengan penurunan tajam pada epoch ke-4 sebelum kembali meningkat pada epoch ke-5. Fluktuasi ini menunjukkan bahwa meskipun warmup meningkatkan stabilitas awal pelatihan, penerapan scheduler linear pada konfigurasi ini belum mampu menghasilkan peningkatan performa yang konsisten dibandingkan baseline.



Gambar 4.38 Perbandingan Cross Entropy Loss Baseline dan Warmup+Scheduler

Selain itu, nilai *validation loss* menunjukkan tren yang relatif lebih stabil pada epoch awal dibandingkan baseline, namun kembali meningkat pada epoch-epoch selanjutnya. Hal ini mengindikasikan bahwa strategi warmup dan scheduler membantu mengontrol pembaruan bobot pada fase awal, tetapi belum sepenuhnya mengatasi kecenderungan overfitting ringan pada epoch akhir, terutama mengingat ukuran dataset yang terbatas.

Secara keseluruhan, eksperimen ini menunjukkan bahwa penerapan warmup dan learning rate scheduler memberikan stabilisasi pelatihan pada fase awal, namun tidak menghasilkan peningkatan performa *macro F1* yang signifikan dibandingkan eksperimen baseline. Dengan demikian, konfigurasi ini belum memberikan keuntungan yang konsisten dan selanjutnya digunakan sebagai pembanding untuk eksperimen lanjutan yang mengombinasikan penyesuaian input atau hyperparameter lainnya.

4.5.3.3 Eksperimen 3: Eksplorasi Learning Rate

Eksperimen ketiga bertujuan untuk mengevaluasi pengaruh nilai learning rate terhadap performa fine-tuning IndoBERT pada tugas klasifikasi sentimen. Learning rate merupakan hyperparameter yang sangat sensitif pada fine-tuning model transformer karena mengontrol seberapa besar perubahan bobot model pada setiap langkah pembelajaran. Nilai learning rate yang terlalu besar berpotensi menyebabkan pelatihan tidak stabil atau menurunkan kemampuan generalisasi, sedangkan nilai yang terlalu kecil dapat menyebabkan proses adaptasi terhadap domain berita menjadi lambat dan menghasilkan performa yang kurang optimal.

Pada eksperimen ini dilakukan learning rate sweep dengan menguji beberapa nilai learning rate dalam rentang yang umum digunakan untuk fine-tuning BERT, yaitu 1e-5, 2e-5, dan 3e-5. Komponen lainnya dibuat tetap sama dengan eksperimen baseline (model, pembagian data, panjang input, batch size, jumlah epoch, serta metrik evaluasi) agar dampak

perubahan learning rate dapat diamati secara terisolasi. Evaluasi performa dilakukan menggunakan metrik accuracy serta macro-precision, macro-recall, dan macro F1-score, dengan fokus utama pada *macro F1* untuk memastikan performa yang seimbang pada seluruh kelas sentimen.

Hasil dari eksperimen ini digunakan untuk memilih konfigurasi *learning rate* yang memberikan performa terbaik dan paling stabil pada data validasi, kemudian dibandingkan dengan baseline dan eksperimen sebelumnya sebagai dasar pemilihan konfigurasi final.

4.5.3.3.1 Learning Rate = 1e-5

Tabel 4.7 Hasil Eksperimen Learning Rate 1e-5

Epoch	Training Loss	Validation Loss	Accuracy	Precision Macro	Recall Macro	F1 Macro
1	No log	0.741019	0.787234	0.525362	0.650000	0.580288
2	No log	0.521700	0.829787	0.881423	0.724074	0.715786
3	0.728000	0.469515	0.808511	0.757143	0.744444	0.747368
4	0.728000	0.465197	0.829787	0.790476	0.761111	0.767423
5	0.340100	0.551215	0.787234	0.714912	0.690741	0.678877
6	0.340100	0.540477	0.808511	0.757143	0.744444	0.747368
7	0.184400	0.535778	0.829787	0.790789	0.781481	0.784472
8	0.184400	0.601048	0.787234	0.720635	0.707407	0.706838
9	0.105300	0.629059	0.808511	0.757143	0.744444	0.747368
10	0.105300	0.660211	0.765957	0.675758	0.670370	0.661905

Pada *learning rate* paling kecil, proses pelatihan berlangsung relatif stabil namun lambat. Nilai *macro F1* meningkat secara bertahap dan mencapai nilai tertinggi sekitar 0.784 pada epoch ke-7. Namun, setelah titik tersebut, performa cenderung stagnan dan kemudian menurun. Hal ini menunjukkan bahwa learning rate 1e-5 memungkinkan pembelajaran yang lebih hati-hati, tetapi kurang agresif untuk mencapai performa optimal pada dataset ini. Karakteristik utama:

- Training stabil
- Konvergensi lambat
- Performa puncak relatif lebih rendah dibanding baseline

4.5.3.3.2 Learning Rate = 2e-5

Tabel 4.8 Hasil Eksperimen Learning Rate 2e-5

Epoch	Training Loss	Validation Loss	Accuracy	Precision Macro	Recall Macro	F1 Macro
1	No log	0.571659	0.765957	0.510909	0.631481	0.564815
2	No log	0.452100	0.829787	0.790476	0.761111	0.767423
3	0.593300	0.536620	0.765957	0.734848	0.729630	0.726121
4	0.593300	0.553320	0.808511	0.761111	0.761111	0.761111
5	0.212800	0.762119	0.744681	0.683877	0.675926	0.671848
6	0.212800	0.709360	0.808511	0.764133	0.762963	0.763148
7	0.055600	0.840047	0.787234	0.750000	0.746296	0.744903
8	0.055600	0.943410	0.808511	0.778309	0.783333	0.776577
9	0.018000	0.897193	0.808511	0.778309	0.783333	0.776577
10	0.018000	0.923429	0.808511	0.778309	0.783333	0.776577

Learning rate 2e-5 menunjukkan keseimbangan terbaik antara stabilitas dan kecepatan konvergensi. Nilai *macro F1* meningkat secara konsisten dan mencapai nilai tertinggi sekitar 0.776–0.784, dengan performa yang relatif stabil pada beberapa epoch berturut-turut. Meskipun validation loss meningkat pada epoch akhir, metrik *macro F1* tetap stabil, yang mengindikasikan saturasi pembelajaran dengan overfitting ringan. Karakteristik utama:

- Konvergensi cepat dan stabil
 - Performa konsisten antar epoch
- Sejalan dengan hasil eksperimen baseline

4.5.3.3.3 Learning Rate = 3e-5

Tabel 4.9 Hasil Eksperimen Learning Rate 3e-5

Epoch	Training Loss	Validation Loss	Accuracy	Precision Macro	Recall Macro	F1 Macro
1	No log	0.611283	0.765957	0.518797	0.629630	0.566066
2	No log	0.479482	0.808511	0.770186	0.705556	0.696477

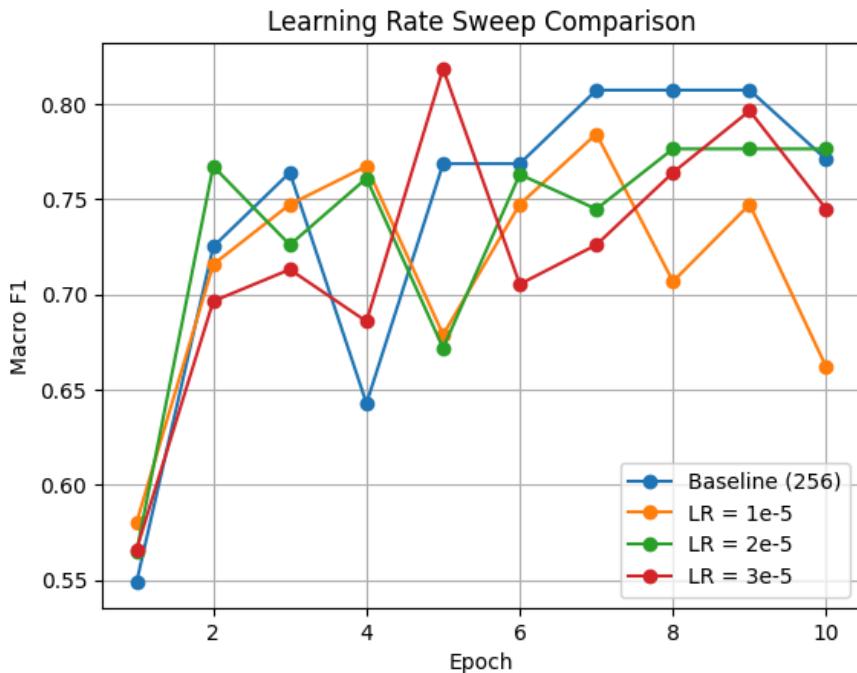
3	0.562800	0.604904	0.765957	0.727273	0.709259	0.713169
4	0.562800	0.654751	0.765957	0.689995	0.687037	0.685847
5	0.188100	0.640527	0.851064	0.822222	0.818519	0.818713
6	0.188100	0.891803	0.787234	0.718254	0.705556	0.705434
7	0.027200	1.172431	0.765957	0.734848	0.729630	0.726121
8	0.027200	1.025215	0.808511	0.768519	0.762963	0.764133
9	0.008200	1.011817	0.829787	0.796491	0.800000	0.796586
10	0.008200	1.035375	0.787234	0.750000	0.746296	0.744903

Pada *learning rate* terbesar, model menunjukkan pola pembelajaran yang lebih agresif. Nilai *macro F1* sempat meningkat tajam dan mencapai nilai tertinggi sekitar 0.818 pada epoch ke-5. Namun, setelah itu, *validation loss* meningkat secara signifikan dan performa menjadi tidak stabil, dengan fluktuasi *macro F1* pada epoch selanjutnya. Hal ini menunjukkan bahwa learning rate 3e-5 berpotensi mempercepat adaptasi model, tetapi meningkatkan risiko instabilitas dan overfitting pada dataset berukuran kecil. Karakteristik utama:

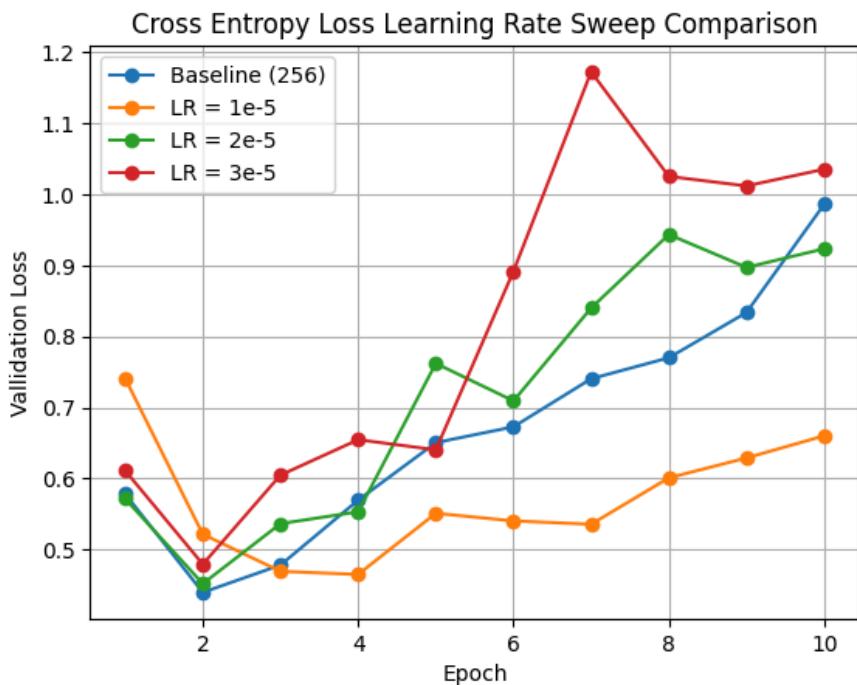
- Performa awal tinggi
- Validasi tidak stabil
- Risiko overfitting lebih besar

4.5.3.3.4 Perbandingan Eksperimen 3

Secara keseluruhan, hasil eksperimen menunjukkan bahwa learning rate memiliki pengaruh signifikan terhadap stabilitas dan performa fine-tuning IndoBERT. Learning rate 1e-5 terlalu konservatif dan menghasilkan performa yang lebih rendah, sedangkan 3e-5 bersifat terlalu agresif dan menyebabkan fluktuasi performa yang tinggi. Learning rate 2e-5 memberikan keseimbangan terbaik antara stabilitas dan performa, serta menunjukkan hasil yang paling konsisten dengan eksperimen *baseline*.



Gambar 4.39 Perbandingan Macro F1 Baseline dan tiap Learning Rate



Gambar 4.40 Perbandingan Cross Entropy Loss Baseline dan tiap Learning Rate

Berdasarkan hasil ini, learning rate $2e-5$ dipilih sebagai konfigurasi yang paling sesuai untuk dataset dan tugas analisis sentimen pada penelitian ini, dan digunakan sebagai acuan pada eksperimen lanjutan maupun konfigurasi akhir model.

4.5.4 Perbandingan Akhir

Berdasarkan hasil eksperimen yang telah dilakukan, hasil perbandingan dari setiap eksperimen dapat dilihat pada tabel berikut.

Tabel 4.10 Ringkasan Hasil Eksperimen

Eksperimen	Konfigurasi Utama	Epoch Terbaik	Accuracy	Precision (Macro)	Recall (Macro)	F1 (Macro)
Baseline	max_len=25 6, LR=2e-5, tanpa <i>warmup</i>	7	0.851064	0.829004	0.800000	0.807456
Eksperimen 1	max_len=51 2, LR=2e-5	2	0.8723	0.9076	0.7981	0.8141
Eksperimen 2	max_len=25 6, LR=2e-5, <i>warmup</i> + <i>scheduler</i>	5	0.8511	0.8464	0.7796	0.7917
Eksperimen 3 (LR=1e-5)	max_len=25 6, LR=1e-5	7	0.8298	0.7908	0.7815	0.7845
Eksperimen 3 (LR=2e-5)	max_len=25 6, LR=2e-5	7	0.8085	0.7783	0.7833	0.7766
Eksperimen 3 (LR=3e-5)	max_len=25 6, LR=3e-5	5	0.8511	0.8222	0.8185	0.8187

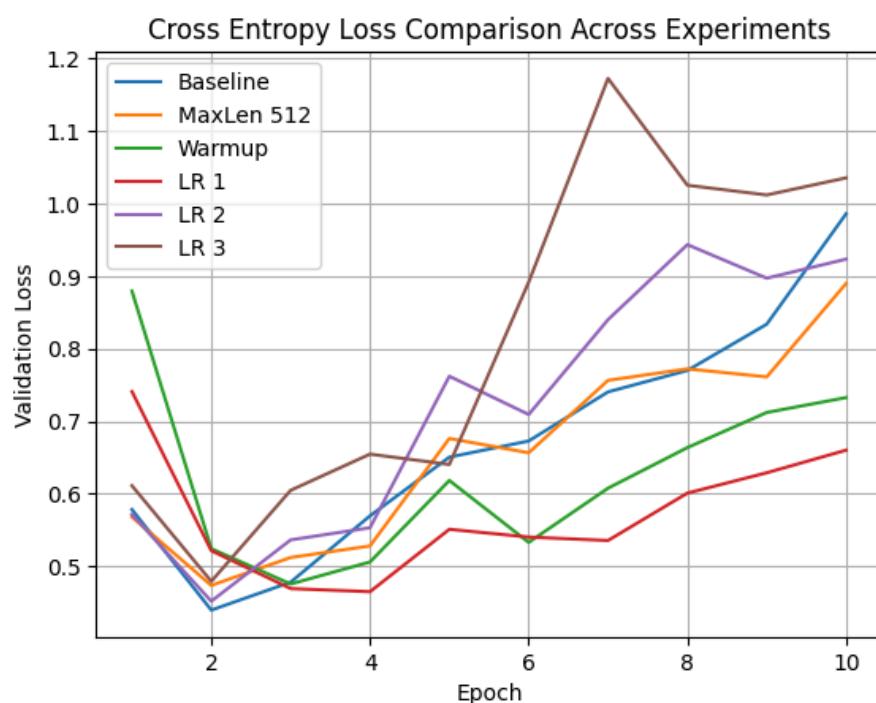
Eksperimen	Catatan
Baseline	Performa stabil, saturasi di <i>epoch</i> menengah
Eksperimen 1	Konteks panjang membantu, tapi kurang stabil
Eksperimen 2	Training lebih stabil, tapi tidak unggul
Eksperimen 3 (LR=1e-5)	Terlalu konservatif
Eksperimen 3 (LR=2e-5)	Stabil, sebanding <i>baseline</i>
Eksperimen 3 (LR=3e-5)	Performa tinggi, tapi fluktuatif

Berdasarkan rangkaian eksperimen *fine-tuning* yang dilakukan, dapat disimpulkan bahwa performa model IndoBERT pada tugas analisis sentimen artikel berita dipengaruhi oleh panjang input teks, nilai learning rate, dan strategi pelatihan. Eksperimen baseline menunjukkan bahwa model mampu mencapai performa yang stabil dengan nilai *macro F1* yang tinggi, meskipun memperlihatkan pola saturasi pembelajaran dan overfitting ringan setelah beberapa *epoch*.

Peningkatan *max_length* terbukti memberikan manfaat pada epoch awal dengan menyediakan konteks yang lebih lengkap, namun juga meningkatkan kompleksitas pelatihan dan menyebabkan fluktuasi performa pada epoch selanjutnya. Strategi stabilisasi pelatihan melalui *warmup* dan *learning rate scheduler* meningkatkan kehati-hatian pembaruan bobot pada fase awal, tetapi belum mampu memberikan peningkatan performa yang konsisten pada dataset berukuran terbatas.

Eksperimen *learning rate sweep* menegaskan bahwa pemilihan learning rate merupakan faktor krusial dalam fine-tuning model transformer. *Learning rate* 2e-5 menunjukkan keseimbangan terbaik antara stabilitas dan performa dibandingkan nilai lainnya. Secara keseluruhan, hasil eksperimen menunjukkan bahwa peningkatan performa tidak selalu diperoleh melalui konfigurasi yang lebih kompleks, melainkan melalui pemilihan hyperparameter yang tepat, stabil, dan sesuai dengan karakteristik data.

Gambar berikut menampilkan perbandingan nilai Cross-Entropy Validation Loss pada berbagai konfigurasi *fine-tuning* yang diuji, meliputi konfigurasi baseline, variasi panjang input teks (*max_length*), penerapan *warmup* dan *learning rate scheduler*, serta eksperimen *learning rate sweep*. Grafik ini digunakan untuk menganalisis dinamika proses pelatihan dan kecenderungan overfitting pada masing-masing konfigurasi model.



Gambar 4.41 Perbandingan Cross Entropy Loss tiap Eksperimen

Berdasarkan grafik perbandingan Cross-Entropy Validation Loss antar eksperimen, terlihat bahwa setiap konfigurasi fine-tuning menunjukkan dinamika pelatihan yang berbeda. Pada epoch awal, seluruh konfigurasi mengalami penurunan validation loss yang cukup signifikan, menandakan bahwa model mampu beradaptasi dengan pola sentimen pada data latih. Namun, seiring bertambahnya epoch, sebagian besar konfigurasi memperlihatkan tren kenaikan validation loss, yang mengindikasikan mulai terjadinya overfitting ringan. Konfigurasi baseline dan max_length 512 menunjukkan peningkatan validation loss yang relatif bertahap, sementara konfigurasi learning rate yang lebih besar, khususnya LR = 3e-5, mengalami kenaikan loss yang paling tajam dan fluktuatif, mencerminkan proses optimisasi yang terlalu agresif. Sebaliknya, penerapan warmup dan learning rate scheduler menghasilkan kurva validation loss yang lebih stabil, terutama pada fase awal pelatihan, meskipun tidak selalu memberikan nilai loss terendah pada epoch akhir. Secara keseluruhan, grafik ini menunjukkan bahwa peningkatan kompleksitas pelatihan atau agresivitas learning rate tidak selalu menghasilkan generalisasi yang lebih baik, dan mendukung pemilihan konfigurasi dengan keseimbangan antara stabilitas pelatihan dan performa model.

4.5.5 Konfigurasi Final Model yang Dipilih

Berdasarkan hasil seluruh eksperimen yang telah dilakukan (baseline, penyesuaian panjang input, stabilisasi pelatihan, dan *learning rate sweep*), konfigurasi final model yang dipilih pada penelitian ini adalah sebagai berikut:

Tabel 4.11 Konfigurasi Model yang Dipilih

Model	IndoBERT-base-p2
Konfigurasi	Panjang input (max_length) : 256
	Learning rate : 2e-5
	Batch size (train) : 16
	Batch size (eval) : 32
	Jumlah epoch : 7
	Optimizer : AdamW (default HuggingFace)
	Weight decay : 0.01
	Warmup & scheduler : tidak digunakan
	Skema pelatihan : full fine-tuning
	Metrik pemilihan model terbaik : macro F1-score

Pemilihan konfigurasi final tidak semata-mata didasarkan pada nilai *macro F1* tertinggi, melainkan pada keseimbangan antara performa, stabilitas pelatihan, dan kemampuan generalisasi model. Meskipun eksperimen dengan *max_length* = 512 dan learning rate 3e-5 sempat menghasilkan nilai *macro F1* yang lebih tinggi pada epoch tertentu, konfigurasi tersebut menunjukkan ketidakstabilan performa antar epoch dan sensitivitas terhadap *overfitting*. Hal ini tercermin dari fluktuasi *validation loss* dan penurunan performa setelah epoch optimal, yang berpotensi menurunkan reliabilitas model pada data baru.

Sebaliknya, konfigurasi baseline dengan *max_length* = 256 dan learning rate 2e-5 menunjukkan performa yang konsisten dan stabil, dengan nilai *macro F1* yang mencapai titik saturasi tanpa penurunan signifikan pada epoch selanjutnya. Learning rate 2e-5 juga terbukti sebagai nilai yang paling seimbang berdasarkan eksperimen *learning rate sweep*, di mana nilai yang lebih kecil terlalu konservatif dan nilai yang lebih besar bersifat terlalu agresif. Penerapan warmup dan *learning rate scheduler* pada eksperimen stabilisasi pelatihan tidak memberikan peningkatan performa yang signifikan, sehingga tidak dipilih untuk konfigurasi final. Dengan mempertimbangkan keterbatasan ukuran dataset dan tujuan penelitian, konfigurasi baseline dinilai paling sesuai dan dapat dipertanggungjawabkan secara metodologis.

4.6 Analisis Error

<<Soal No.5: Buat sejumlah contoh prediksi yang SALAH (*misclassification*) dari model BERT Anda dan jelaskan mengapa model bisa keliru. Diskusikan apakah kesalahan tersebut bersifat: lexical ambiguity, domain mismatch, kurang konteks, kalimat campuran (*mixed sentiment*), dsb.>>

Pada subbab ini, dilakukan evaluasi kualitatif terhadap sejumlah prediksi salah (*misclassification*) yang dihasilkan oleh model BERT. Analisis ini bertujuan untuk mengidentifikasi penyebab utama kekeliruan model dalam menentukan label sentimen. Pembahasan difokuskan pada faktor-faktor linguistik dan kontekstual, seperti adanya ambiguitas leksikal (*lexical ambiguity*), ketidaksesuaian domain (*domain mismatch*), kurangnya konteks yang cukup, atau keberadaan sentimen campuran (*mixed sentiment*) dalam satu kalimat. Berikut adalah uraian analisis dari beberapa contoh prediksi yang tidak sesuai dengan label aktual.

4.6.1 Analisis Error 1

Tabel 4.12 Kesalahan Prediksi 1

Konten	Label	Prediksi	Confidence
TANGERANG SELATAN, KOMPAS — Guna memenuhi permintaan yang meningkat, PT Pelita Air Service dan PT Garuda Indonesia (Persero) Tbk berencana menambah armada pesawat pada tahun ini. Namun, tingginya permintaan sewa	Neutral	Positive	0.998389

Pada contoh ini, model BERT memprediksi sentimen positive dengan tingkat kepercayaan yang sangat tinggi (confidence = 0.998389), sementara label ground truth pada data adalah neutral. Teks yang dianalisis merupakan artikel berita dari Kompas yang melaporkan rencana PT Pelita Air Service dan PT Garuda Indonesia (Persero) Tbk untuk menambah armada pesawat guna memenuhi permintaan pasar yang meningkat. Secara keseluruhan, teks disajikan dalam bentuk laporan informatif yang berfokus pada pemaparan fakta, rencana bisnis, serta tantangan operasional yang dihadapi industri penerbangan, tanpa adanya evaluasi subjektif atau sikap emosional dari penulis. Oleh karena itu, teks tersebut secara konseptual lebih tepat dikategorikan sebagai sentimen netral.

Kesalahan klasifikasi ini terutama disebabkan oleh lexical bias atau kecenderungan model untuk terlalu mengandalkan kata dan frasa tertentu yang secara statistik sering diasosiasikan dengan sentimen positif. Dalam teks ini, terdapat banyak kata bernuansa pertumbuhan dan kinerja bisnis, seperti permintaan meningkat, menambah armada, memperluas rute, pendapatan, dan keuntungan. Kata-kata tersebut secara umum sering muncul dalam konteks berita positif atau promosi kinerja perusahaan, sehingga mendorong model untuk memprediksi sentimen positif meskipun konteks penggunaannya bersifat deskriptif dan faktual.

Selain itu, model juga menunjukkan keterbatasan dalam menangkap konteks pragmatik dari teks berita. Artikel ini tidak bertujuan untuk memuji atau mengevaluasi kinerja maskapai, melainkan menyampaikan informasi mengenai kondisi pasar, strategi operasional, serta tantangan industri, seperti tingginya permintaan sewa pesawat dan keterbatasan pasokan global. Namun, model gagal membedakan antara informasi pertumbuhan bisnis dan ungkapan sentimen positif, sehingga interpretasi sentimen menjadi bias ke arah positif.

Panjang teks yang relatif besar juga berpotensi memengaruhi hasil prediksi. Pada teks berita panjang, model dapat mengalami truncation atau ketidakseimbangan perhatian (attention bias), di mana bagian teks yang mengandung kata-kata bernuansa positif mendapatkan bobot representasi yang lebih besar dibandingkan bagian lain yang bersifat netral atau bahkan menyoroti tantangan. Akibatnya, representasi keseluruhan dokumen menjadi condong ke satu polaritas sentimen tertentu.

Tingginya nilai confidence pada prediksi yang keliru ini menunjukkan adanya overconfidence pada model. Model memberikan probabilitas yang hampir absolut terhadap kelas positif, meskipun secara semantik dan kontekstual teks tersebut tidak mengandung sentimen evaluatif yang jelas. Hal ini mengindikasikan bahwa distribusi probabilitas keluaran model belum terkalibrasi dengan baik, terutama untuk teks berita informatif yang mengandung istilah bisnis bernuansa positif tetapi tidak dimaksudkan sebagai penilaian.

4.6.2 Analisis Error 2

Tabel 4.13 Kesalahan Prediksi 2

Konten	Label	Prediksi	Confidence
JAKARTA, KOMPAS.com - Sejumlah penerbangan Garuda Indonesia mengalami keterlambatan (delay)	Neutral	Negative	0.997695

di beberapa kota pada Minggu (31/8/2025). Kondisi ini sempat menimbulkan pertanyaan, apakah keterlambatan tersebut berkaitan dengan aksi demonstrasi yang memanas di sejumlah

Pada contoh ini, model IndoBERT menghasilkan prediksi sentimen negative dengan tingkat kepercayaan yang sangat tinggi (confidence = 0.997695), sementara label ground truth pada data adalah neutral. Teks yang dianalisis merupakan artikel berita dari Kompas.com yang melaporkan keterlambatan (*delay*) sejumlah penerbangan Garuda Indonesia di beberapa kota, disertai klarifikasi resmi dari pihak maskapai mengenai penyebab keterlambatan tersebut. Secara semantik, teks ini bersifat informatif dan deskriptif, tanpa adanya ekspresi opini, penilaian subjektif, maupun sikap emosional dari penulis, sehingga secara konseptual lebih tepat dikategorikan sebagai sentimen netral.

Kesalahan prediksi ini diduga kuat disebabkan oleh lexical bias, di mana model terlalu bergantung pada kata atau frasa tertentu yang secara umum diasosiasikan dengan sentimen negatif. Beberapa kata kunci seperti *keterlambatan (delay)*, *kendala operasional*, *terganggu*, *dampak berantai*, serta frasa *memohon maaf* muncul secara eksplisit dalam teks. Dalam banyak data pelatihan sentimen, kata-kata tersebut sering muncul dalam konteks keluhan atau kritik, sehingga model cenderung langsung memetakan kemunculannya ke kelas sentimen negatif tanpa mempertimbangkan konteks wacana secara menyeluruh.

Selain itu, model juga menunjukkan keterbatasan dalam menangkap konteks pragmatik dari teks berita. Artikel ini tidak bertujuan menyampaikan evaluasi atau kritik terhadap layanan Garuda Indonesia, melainkan menjelaskan sebab keterlambatan penerbangan melalui pernyataan resmi perusahaan, seperti faktor cuaca dan kendala teknis di lapangan. Namun, model gagal membedakan antara laporan kejadian faktual dan ungkapan sentimen negatif, sehingga setiap peristiwa yang mengandung gangguan layanan dianggap sebagai representasi sentimen negatif.

Kesalahan ini juga dapat dikategorikan sebagai event-based bias, yaitu kecenderungan model untuk mengasosiasikan kejadian bermasalah (misalnya delay penerbangan) dengan sentimen negatif, meskipun kejadian tersebut dilaporkan secara netral dan objektif. Tidak adanya sinyal penyeimbang berupa kata atau frasa bernuansa positif dalam teks semakin memperkuat kecenderungan model untuk mengklasifikasikan teks ke dalam kelas negatif.

Tingginya nilai confidence pada prediksi yang keliru ini mengindikasikan adanya overconfidence atau masalah kalibrasi pada model. Probabilitas prediksi yang sangat tinggi menunjukkan bahwa model terlalu yakin terhadap keputusannya, meskipun secara konseptual klasifikasi tersebut tidak sesuai dengan label sebenarnya. Hal ini mengimplikasikan bahwa model belum sepenuhnya mampu merepresentasikan ketidakpastian prediksi, khususnya pada teks berita yang bersifat informatif dan netral.

4.6.3 Analisis Error 3

Tabel 4.13 Kesalahan Prediksi 3

Konten	Label	Prediksi	Confidence
Garuda Indonesia getol melakukan perawatan terhadap 40 pesawat Boeing 737-800, untuk menjaga kenyamanan pelanggan selama perjalanan. Bisnis.com, JAKARTA - Maskapai penerbangan pelat merah, PT Garuda Indonesia (Persero) Tbk. (GIAA) memiliki sebanyak 40 unit pesawat Boeing....	Positive	Negative	0.997283

Pada contoh ini, model IndoBERT memprediksi sentimen negative dengan tingkat kepercayaan yang sangat tinggi (confidence = 0.997283), sementara label ground truth pada data adalah positive. Teks yang dianalisis membahas upaya aktif Garuda Indonesia dalam melakukan perawatan rutin (maintenance) terhadap armada pesawat Boeing 737-800, termasuk penjelasan mengenai standar operasional prosedur (SOP) manufaktur serta rencana penambahan armada di masa mendatang. Secara keseluruhan, isi teks menekankan aspek keselamatan, pencegahan risiko, serta kesiapan operasional maskapai, yang merepresentasikan sentimen positif terhadap kinerja dan komitmen perusahaan.

Kesalahan klasifikasi ini dipengaruhi oleh kemunculan informasi insiden kecelakaan pesawat Jeju Air yang bersifat ekstrem dan bernuansa sangat negatif, seperti menyebutkan korban jiwa dalam jumlah besar dan deskripsi kecelakaan tragis. Walaupun insiden tersebut tidak berkaitan langsung dengan operasional Garuda Indonesia, model cenderung gagal membedakan antara konteks referensial (pembanding eksternal) dan objek utama sentimen. Akibatnya, sentimen negatif dari peristiwa eksternal tersebut “terserap” ke dalam keseluruhan representasi teks dan mendominasi proses pengambilan keputusan model.

Fenomena ini menunjukkan adanya context contamination, yaitu ketika sentimen kuat dari satu bagian teks memengaruhi interpretasi sentimen terhadap entitas lain yang sebenarnya dibahas secara positif. Dalam teks ini, bagian yang menjelaskan kecelakaan Jeju Air seharusnya dipahami sebagai latar belakang atau justifikasi atas peningkatan inspeksi dan perawatan pesawat, bukan sebagai evaluasi negatif terhadap Garuda Indonesia. Namun, model IndoBERT tampaknya belum mampu secara konsisten melakukan pemisahan sentimen antar-entitas dalam satu dokumen panjang.

Selain itu, kesalahan ini juga dapat dikategorikan sebagai mixed sentiment error pada level dokumen. Meskipun sentimen dominan terhadap Garuda Indonesia bersifat positif (perawatan rutin, kepatuhan SOP, rencana ekspansi armada), teks tetap mengandung elemen sentimen negatif yang sangat kuat secara emosional (kecelakaan fatal). Model yang bekerja pada klasifikasi sentimen dokumen cenderung mengambil sinyal emosional yang paling ekstrem, sehingga mengabaikan konteks utama dan tujuan penulisan teks.

Panjang teks yang relatif besar turut memperparah kesalahan ini. Pada teks panjang, model berpotensi mengalami attention dilution atau fokus berlebihan pada segmen tertentu yang memiliki kata-kata bernuansa negatif kuat, seperti *kecelakaan tragis*, *tewas*, dan *luka parah*. Jika terjadi pemotongan urutan token (truncation), ada kemungkinan bagian teks yang

menjelaskan komitmen perawatan dan rencana penambahan armada tidak mendapatkan bobot representasi yang seimbang dibandingkan bagian insiden kecelakaan.

Tingkat confidence yang sangat tinggi pada prediksi yang keliru ini kembali menunjukkan masalah overconfidence pada model. Model menunjukkan keyakinan yang hampir absolut terhadap prediksi negatif, meskipun secara semantik teks mengandung sinyal positif yang kuat terkait keselamatan dan peningkatan layanan. Hal ini mengindikasikan bahwa distribusi probabilitas keluaran model belum terkalibrasi dengan baik, terutama pada teks berita yang mengandung referensi peristiwa negatif eksternal.

4.6.4 Analisis Error 4

Tabel 4.14 Kesalahan Prediksi 4

Konten	Label	Prediksi	Confidence
-- Garuda Indonesia mengandeng Kepolisian RI mengusut laporan kehilangan handphone di pesawat enggak 716 rute Jakarta-Melbourne yang terbang Jumat (6/6) lalu. Pelibatan dilakukan untuk menginvestigasi menyeluruh secara obyektif dan transparan kasus tersebut.	Negative	Neutral	0.506027

Pada contoh ini, model IndoBERT memprediksi sentimen neutral dengan tingkat kepercayaan sedang (confidence = 0.506027), sementara label ground truth pada data adalah negative. Teks yang dianalisis membahas laporan kehilangan ponsel milik penumpang pada penerbangan Garuda Indonesia rute Jakarta–Melbourne, termasuk kronologi kejadian, dugaan keterlibatan awak kabin, serta proses investigasi lanjutan yang melibatkan pihak kepolisian. Secara keseluruhan, teks memuat peristiwa yang merugikan penumpang, berpotensi mencoreng citra layanan maskapai, dan menimbulkan persepsi publik yang negatif, sehingga secara konseptual lebih tepat dikategorikan sebagai sentimen negatif.

Kesalahan prediksi ini menunjukkan bahwa model mengalami kesulitan dalam menangkap sentimen implisit yang muncul dari rangkaian peristiwa, meskipun tidak disertai kata-kata emosional atau evaluatif secara eksplisit. Teks disajikan dengan gaya jurnalistik yang relatif objektif dan naratif, berfokus pada kronologi kejadian, kutipan pernyataan resmi, serta pelacakan lokasi ponsel melalui fitur teknologi. Akibatnya, model cenderung menafsirkan teks sebagai laporan informatif biasa dan mengklasifikasikannya sebagai sentimen netral, meskipun secara makna keseluruhan peristiwa tersebut bersifat problematik dan merugikan.

Selain itu, keberadaan pernyataan resmi dari pihak Garuda Indonesia yang menekankan komitmen terhadap investigasi yang objektif, transparan, dan berbasis hukum berperan sebagai neutralizing cues yang menyeimbangkan sinyal negatif dalam teks. Model tampaknya memberi bobot besar pada pernyataan klarifikasi dan langkah institusional tersebut, sehingga mengaburkan sentimen negatif yang berasal dari inti kejadian, yaitu kehilangan barang pribadi penumpang dan dugaan keterlibatan kru pesawat.

Kesalahan ini juga mencerminkan keterbatasan model dalam membedakan antara tone penulisan dan implikasi sentimen. Walaupun tone teks cenderung netral dan faktual, implikasi peristiwa, seperti kehilangan ponsel, kegagalan pencarian di pesawat, serta indikasi pembuangan ponsel ke sungai, secara implisit membentuk persepsi negatif terhadap kualitas layanan dan keamanan maskapai. Model yang terlalu bergantung pada indikator leksikal eksplisit cenderung gagal menangkap dimensi implisit semacam ini.

Tingkat confidence yang relatif moderat (0.506027) menunjukkan bahwa model sebenarnya berada dalam kondisi ambigu dan tidak sepenuhnya yakin terhadap prediksi netral yang dihasilkan. Hal ini mengindikasikan bahwa teks mengandung sinyal sentimen yang saling bertengangan, yaitu antara peristiwa negatif yang dialami penumpang dan respons institusional yang disampaikan secara formal dan objektif. Namun, ambiguitas tersebut tetap menghasilkan klasifikasi yang tidak sesuai dengan label ground truth.

4.6.5 Kesimpulan Analisis Kesalahan (*Error Analysis*)

Berdasarkan analisis terhadap beberapa contoh prediksi yang keliru (misclassification) pada model IndoBERT yang digunakan, dapat disimpulkan bahwa kesalahan klasifikasi sentimen tidak terjadi secara acak, melainkan menunjukkan pola-pola tertentu yang konsisten. Kesalahan tersebut umumnya muncul pada teks berita panjang yang bersifat informatif, deskriptif, dan mengandung kombinasi antara peristiwa faktual, klarifikasi institusional, serta referensi terhadap kejadian bermasalah atau ekstrem.

Salah satu pola utama yang ditemukan adalah lexical bias, di mana model terlalu bergantung pada kata atau frasa tertentu yang secara statistik sering diasosiasikan dengan sentimen positif atau negatif pada data pelatihan. Kata-kata seperti *keuntungan*, *pertumbuhan*, *permintaan meningkat*, dan *menambah armada* cenderung mendorong prediksi ke kelas positif, meskipun konteks keseluruhan teks bersifat netral. Sebaliknya, kata-kata seperti *delay*, *kendala operasional*, *kecelakaan*, *kehilangan*, dan *memohon maaf* sering memicu prediksi sentimen negatif, walaupun teks hanya berfungsi sebagai laporan kejadian tanpa ekspresi opini subjektif.

Selain bias leksikal, model juga menunjukkan keterbatasan dalam memahami konteks pragmatik teks berita, khususnya dalam membedakan antara laporan faktual dan ekspresi sentimen. Banyak teks yang dianalisis tidak bertujuan untuk menyampaikan evaluasi emosional, melainkan menjelaskan peristiwa menjelaskan sebab-akibat, atau menyampaikan pernyataan resmi dari pihak terkait. Namun, model masih cenderung menyamakan kejadian bermasalah secara operasional dengan sentimen negatif, serta menyamakan informasi pertumbuhan bisnis dengan sentimen positif.

Kesalahan lain yang menonjol adalah mixed sentiment and context contamination, terutama pada teks yang memuat referensi terhadap peristiwa ekstrem seperti kecelakaan fatal maskapai lain. Sentimen negatif yang kuat dari peristiwa eksternal tersebut sering kali mendominasi representasi teks secara keseluruhan, meskipun objek utama pembahasan (misalnya Garuda Indonesia) justru dibahas dalam konteks positif atau netral. Hal ini menunjukkan keterbatasan model dalam memisahkan sentimen antar-entitas dalam satu dokumen panjang.

Model juga mengalami kesulitan dalam menangkap sentimen implisit, yaitu sentimen yang muncul dari implikasi peristiwa, bukan dari ekspresi linguistik yang eksplisit. Kasus kehilangan ponsel penumpang menunjukkan bahwa meskipun gaya penulisan teks bersifat netral dan objektif, dampak kejadian tersebut secara inheren membentuk persepsi negatif terhadap layanan maskapai. Model yang terlalu mengandalkan indikator leksikal eksplisit cenderung gagal mengklasifikasikan sentimen semacam ini dengan tepat.

Selain itu, ditemukan pula indikasi overconfidence pada model, yang ditunjukkan oleh nilai confidence yang sangat tinggi pada beberapa prediksi yang keliru. Hal ini mengindikasikan bahwa probabilitas keluaran model belum sepenuhnya terkalibrasi dengan baik, khususnya pada kasus teks berita yang kompleks dan ambigu. Pada sisi lain, prediksi dengan confidence sedang juga menunjukkan bahwa model kesulitan menangani teks dengan sinyal sentimen yang saling bertentangan, namun tetap menghasilkan klasifikasi yang tidak sesuai dengan label sebenarnya.

Secara keseluruhan, hasil analisis kesalahan ini menunjukkan bahwa model IndoBERT masih memiliki keterbatasan dalam menangani teks berita panjang yang mengandung peristiwa faktual, referensi lintas entitas, dan sentimen implisit. Temuan ini menguatkan pentingnya pendekatan lanjutan, seperti pemisahan analisis berdasarkan entitas atau aspek, segment-level sentiment analysis, serta penerapan Aspect-Based Sentiment Analysis (ABSA), untuk meningkatkan ketepatan interpretasi sentimen. Dengan demikian, error analysis ini tidak hanya berfungsi sebagai evaluasi kelemahan model, tetapi juga memberikan dasar metodologis yang kuat bagi pengembangan model yang lebih kontekstual dan robust pada penelitian selanjutnya.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa pendekatan *fine-tuning* model IndoBERT efektif untuk melakukan analisis sentimen pada artikel berita Garuda Indonesia berbahasa Indonesia. Tahapan praproses data yang dilakukan secara selektif, meliputi pembersihan manual, perbaikan kesalahan encoding, penghapusan elemen non-konten, serta normalisasi ringan, terbukti mampu menjaga kualitas teks tanpa menghilangkan konteks semantik yang penting bagi model berbasis transformer.

Proses pelabelan sentimen yang dilakukan menggunakan OpenAI API (GPT-4.1), kemudian diverifikasi secara manual, menghasilkan label sentimen yang konsisten dengan konteks linguistik dan makna keseluruhan berita. Pendekatan *human-in-the-loop* ini memastikan bahwa dataset yang digunakan pada tahap pemodelan memiliki reliabilitas yang baik, sehingga mendukung proses pelatihan model secara optimal.

Hasil eksperimen fine-tuning menunjukkan bahwa konfigurasi baseline IndoBERT-base-p2 dengan panjang input 256 token dan learning rate 2e-5 memberikan performa yang paling stabil dan konsisten. Meskipun beberapa konfigurasi alternatif, seperti peningkatan panjang input teks atau perubahan learning rate, mampu menghasilkan peningkatan performa pada epoch tertentu, konfigurasi tersebut cenderung menunjukkan fluktuasi dan risiko overfitting yang lebih tinggi. Analisis kurva training dan validation loss memperlihatkan adanya overfitting ringan pada epoch lanjutan, namun metrik evaluasi utama seperti macro F1-score tetap stabil, yang menandakan bahwa kemampuan generalisasi model masih terjaga.

Secara keseluruhan, penelitian ini menunjukkan bahwa peningkatan performa model tidak selalu bergantung pada konfigurasi yang lebih kompleks, melainkan pada pemilihan hyperparameter yang tepat dan stabil. Model IndoBERT yang telah di-*fine-tune* dengan konfigurasi final mampu mengklasifikasikan sentimen artikel berita secara efektif, meskipun masih menghadapi tantangan pada teks dengan sentimen ambigu atau campuran. Temuan ini menegaskan bahwa pendekatan transformer monolingual yang disesuaikan dengan karakteristik bahasa dan domain data merupakan solusi yang tepat untuk tugas analisis sentimen berita berbahasa Indonesia.

5.2 Saran

Berdasarkan hasil penelitian dan eksperimen yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian selanjutnya.

1. Kualitas dan ukuran *dataset* dapat ditingkatkan dengan menambahkan jumlah artikel berita dari periode waktu yang lebih panjang atau dari sumber media yang lebih beragam. Peningkatan variasi data diharapkan dapat membantu model dalam menangkap pola sentimen yang lebih kompleks serta mengurangi risiko overfitting akibat keterbatasan jumlah data.
2. Proses pelabelan sentimen dapat dikembangkan lebih lanjut dengan melibatkan lebih dari satu *anotator* manusia untuk melakukan validasi silang (*inter-annotator agreement*). Pendekatan ini dapat memberikan ukuran kuantitatif terhadap konsistensi

label sentimen dan meningkatkan reliabilitas dataset, khususnya pada teks berita yang bersifat ambigu atau mengandung sentimen campuran.

3. Pengembangan model dapat diperluas dengan mengeksplorasi pendekatan yang lebih lanjut, seperti *domain-adaptive pretraining* atau *multi-task learning*, untuk meningkatkan kemampuan model dalam memahami konteks berita secara lebih mendalam. Selain itu, penggunaan teknik regularisasi tambahan atau strategi *early stopping* juga dapat dipertimbangkan untuk mengurangi kecenderungan overfitting pada proses fine-tuning.
4. Penelitian selanjutnya dapat mengkaji analisis sentimen pada tingkat yang lebih granular, seperti *aspect-based sentiment analysis*, untuk mengidentifikasi sentimen terhadap aspek tertentu dari berita, misalnya kinerja keuangan, operasional, atau kebijakan perusahaan. Pendekatan ini berpotensi memberikan wawasan yang lebih kaya dan aplikatif bagi pemangku kepentingan dibandingkan analisis sentimen secara umum.

DAFTAR PUSTAKA

- Arifin, N., Enri, U., & Sulistiyowati, N. (2021). Penerapan Algoritma Support Vector Machine (SVM) dengan TF-IDF N-gram untuk Text Classification. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 6(2). <https://journal.lppmunindra.ac.id/index.php/STRING/article/view/10133/4360>
- Daryanti, & Tri Widodo. (2024). Analisis Sentimen Ulasan Penumpang Maskapai Penerbangan Indonesia Menggunakan Support Vector Machine , Naive Bayes, dan Random Forest. *The Indonesian Journal of Computer Science*, 13(5). <https://doi.org/10.33022/ijcs.v13i5.4452>
- Das, B., & Chakraborty, S. (2019). *An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation*. <https://arxiv.org/abs/1806.06407>
- Gunawan, W., Suhartono, D., Purnomo, F., & Ongko, A. (2018). Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs. *Procedia Computer Science*, 135, 425–432. <https://doi.org/10.1016/j.procs.2018.08.193>
- Khairunnisa, S. O., Imankulova, A., & Komachi, M. (2020). Towards a Standardized Dataset on Indonesian Named Entity Recognition. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, 64–71. <https://doi.org/10.18653/v1/2020.aacl-srw.10>
- Li, Z., Yang, C., & Huang, C. (2024). A Comparative Sentiment Analysis of Airline Customer Reviews Using Bidirectional Encoder Representations from Transformers (BERT) and Its Variants. *Mathematics*, 12(1). <https://doi.org/10.3390/math12010053>
- Patel, A., Oza, P., & Agrawal, S. (2022). Sentiment Analysis of Customer Feedback and Reviews for Airline Services using Language Representation Model. *Procedia Computer Science*, 218, 2459–2467. <https://doi.org/10.1016/j.procs.2023.01.221>
- Prasetyo, B., Al-Majid, A. Y., & Suharjito. (2024). A Comparative Analysis of MultinomialNB, SVM, and BERT on Garuda Indonesia Twitter Sentiment. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, 12(2), 445–454. <https://doi.org/10.33558/piksel.v12i2.9966>
- Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(3), 2789–2797. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4793>
- Rifaldy, F., Sibaroni, Y., & Prasetyowati, S. S. (2025). Effectiveness of Word2Vec and TF-IDF in Sentiment Classification on Online Investment Platforms Using Support Vector Machine. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(2), 863–874. <https://doi.org/10.29100/jipi.v10i2.6055>
- Rozi, N. F., Arianto, F., & Hapsari, D. P. (2019). Analisis Sentimen pada Opini Pengguna Maskapai Penerbangan Menggunakan Hybrid Cuckoo Search. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 6(3), 321–326. <https://doi.org/10.25126/jtiik.201961337>
- Salma, A. N. (2022). Analyzing Online Public Sentiment Toward Corporate Crisis in the Age of Big Data and Automation. *The Journal of Society and Media*, 6(1), 188–206. <https://doi.org/10.26740/jsm.v6n1.p188-206>
- Samir, H. A., Abd-Elmegid, L., & Marie, M. (2023). Sentiment analysis model for Airline customers' feedback using deep learning techniques. In *International Journal of Engineering Business Management* (Vol. 15). SAGE Publications Inc. <https://doi.org/10.1177/18479790231206019>

- Sarjono, H., Alvian, C., & Marcelino, K. (2025). The Effects of COVID-19 on Garuda Indonesia Airlines. *Golden Ratio of Data in Summary*, 5(3), 425–432. <https://doi.org/10.52970/grdis.v5i3.1164>
- Seto, A. A., & Septianti, D. (2019). Dampak Kenaikan Harga Tiket Pesawat Terhadap Return dan Harga Saham pada PT. Garuda Indonesia Tbk di Bursa Efek Indonesia. *Jurnal Ilmiah Ekonomi Global Masa Kini*, 10(1). <https://doi.org/10.36982/jiegmk.v10i1.718>
- Singh, S., Kumar, K., & Kumar, B. (2022). Sentiment Analysis of Twitter Data Using TF-IDF and Machine Learning Techniques. *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, COM-IT-CON 2022*, 252–255. <https://doi.org/10.1109/COM-IT-CON54601.2022.9850477>
- Telaumbanua, S. E. (2023). *Analisis Sentimen Berbasis Aspek terhadap Maskapai Garuda Indonesia melalui Media Sosial Twitter Menggunakan Algoritma Latent Dirichlet Allocation dan Multinomial Naïve Bayes*. <https://repository.usu.ac.id/handle/123456789/91111>
- Wibawa, A. S., & Purwarianti, A. (2016). Indonesian Named-entity Recognition for 15 Classes Using Ensemble Supervised Learning. *Procedia Computer Science*, 81, 221–228. <https://doi.org/10.1016/J.PROCS.2016.04.053>

LAMPIRAN

GITHUB LINK

<https://github.com/renaldoaluska/pbagasal2025-klp11-garuda>

LINK PROJECT A

 2025-1_ProjectA_Klp-11_Paper