
ANALISIS SENTIMEN MENGGUNAKAN MODEL CONCATENATION CLASSIFICATION INDOBERT UNTUK ULASAN PENGGUNA APLIKASI MYTELKOMSEL DI PLAYSTORE

Tri Fitria Ningsih, S.Kom., M.Kom.¹⁾, Dr. Ronsen Purba, M.Sc.²⁾, Fermi Pasha, B.Sc., M.Sc., PhD.³⁾
Teknologi Informasi
Universitas Mikroskil

Jl. M.H Thamrin No.140, Pusat Ps., Kec. Medan Kota, Kota Medan, Sumatera Utara 20212
e-mail: tri.fitria@students.mikroskil.ac.id¹⁾, ronsen@mikroskil.ac.id²⁾, muhammad.pasha@mikroskil.ac.id³⁾

Abstract

In the digital era, user reviews of mobile applications have become an important data source for evaluating and improving service quality. This study aims to analyze user sentiment toward the MyTelkomsel application on Google PlayStore using the Concatenation Classification IndoBERT approach, which combines embeddings from IndoBERT and FastText to enrich the semantic representation of text. A total of 30,000 user reviews in Indonesian from 2022 to 2024 were collected through web scraping using the google-play-scraper library. The data was processed through several preprocessing stages, including normalization, cleaning, stopword removal, and stemming, followed by sentiment labeling into five categories: very negative, negative, neutral, positive, and very positive, based on polarity scores. Modeling was performed by combining the [CLS] token vector from IndoBERT (768 dimensions) and the FastText vector (300 dimensions), resulting in a 1068-dimensional vector. The dataset was split into 80% training data, 20% testing data and 80% training data, 10% testing data and 10% validation data. Evaluation using metrics such as accuracy, precision, recall, and F1-score showed high performance, with accuracy reaching 95% in 2022–2023 and increasing to 96% in 2024. These results indicate that the IndoBERT concatenation approach significantly improves sentiment classification accuracy and is effective in handling unstructured user review texts

Keywords: Sentiment analysis, IndoBERT, Concatenation Classification, FastText, MyTelkomsel, NLP, Google PlayStore

1. Pendahuluan

Dalam era digital saat ini, aplikasi mobile telah menjadi bagian tak terpisahkan dari kehidupan masyarakat. Aplikasi layanan telekomunikasi, seperti MyTelkomsel, memungkinkan pengguna untuk mengakses berbagai layanan, mulai dari pembelian paket data hingga pengelolaan akun. Dengan semakin banyaknya pengguna, ulasan yang diberikan di platform seperti PlayStore menjadi sumber data berharga bagi perusahaan dalam memahami pengalaman pengguna dan meningkatkan kualitas layanan. Namun, banyaknya ulasan yang tidak terstruktur dan bersifat subjektif membuat analisis manual menjadi tidak efisien, sehingga diperlukan pendekatan berbasis kecerdasan buatan untuk mengolah data secara otomatis dan akurat.

Tren terkini menunjukkan bahwa model berbasis Transformer, seperti IndoBERT, semakin banyak digunakan dalam pemrosesan bahasa alami (NLP) untuk bahasa Indonesia. Studi terbaru oleh Irmawan et al. (2024) menunjukkan bahwa IndoBERT mampu meningkatkan akurasi analisis sentimen dalam ulasan aplikasi perjalanan di Indonesia, dibandingkan dengan metode tradisional seperti Support Vector Machine atau Naïve Bayes [1]. Selain itu, penelitian oleh Fiarni (2024) membuktikan bahwa IndoBERT yang di fine-tune dapat menangani analisis berbasis aspek dalam ulasan aplikasi video streaming secara lebih efektif dibandingkan metode lain [2]. Keunggulan IndoBERT dalam memahami bahasa Indonesia secara kontekstual membuatnya menjadi pilihan yang potensial dalam menganalisis ulasan pengguna secara lebih akurat.

Meskipun IndoBERT telah digunakan dalam berbagai penelitian, masih terdapat kesenjangan dalam penerapannya pada analisis sentimen ulasan pengguna aplikasi layanan telekomunikasi. Sebagian besar studi yang ada lebih berfokus pada aplikasi e-commerce, transportasi, atau layanan kesehatan (Naufal, 2025) [3]. Selain itu, metode yang umum digunakan dalam klasifikasi sentimen masih terbatas pada pendekatan konvensional tanpa optimalisasi arsitektur model untuk meningkatkan performa analisis.

Salah satu pendekatan yang berpotensi meningkatkan akurasi analisis sentimen adalah Concatenation Classification, yaitu metode yang menggabungkan fitur dari berbagai lapisan IndoBERT untuk memperoleh pemahaman sentimen yang lebih mendalam. Dengan mengadopsi metode ini, diharapkan dapat menangkap lebih banyak informasi dari teks ulasan, sehingga menghasilkan klasifikasi yang lebih akurat dibandingkan pendekatan tradisional. Namun, penelitian mengenai efektivitas metode ini dalam analisis sentimen masih terbatas, khususnya dalam konteks ulasan aplikasi layanan telekomunikasi.

Secara praktis, hasil penelitian diharapkan dapat membantu pengembang aplikasi dalam memahami kebutuhan pengguna secara lebih akurat, sehingga dapat meningkatkan kualitas layanan berdasarkan opini pengguna yang dianalisis secara sistematis dan berbasis data melalui sebuah penelitian yang berjudul “Analisis Sentimen Menggunakan Model Concatenation Classification IndoBERT Untuk Ulasan Pengguna Aplikasi MyTelkomsel di Playstore”

2. Landasan Teori

Pada bagian ini berisikan landasan teori dan akan dijelaskan tinjauan pustaka yang berkaitan dengan penelitian yang dilakukan.

Analisis Sentimen

Analisis sentimen adalah teknik pemrosesan data alami atau Natural Language Processing (NLP) yang digunakan untuk mengklasifikasi opini dalam kategori sentimen positif, negatif atau netral [4]. Metode ini biasanya diterapkan pada ulasan produk media sosial dan komentar pengguna untuk memahami persepsi terhadap layanan atau produk tertentu. Dalam konteks aplikasi MyTelkomsel, analisis sentimen berguna untuk memahami persepsi pelanggan terhadap fitur dan layanan yang disediakan, yang kemudian dapat digunakan sebagai dasar untuk meningkatkan kualitas produk. Teknik analisis sentimen secara umum dibagi menjadi tiga, yaitu:

1. Pendekatan Berbasis Leksikon.

Pada pendekatan ini kata-kata dalam bentuk teks dicocokkan dengan daftar kata yang telah dikategorikan berdasarkan maknanya. Contohnya adalah leksikon sentimen yang berisi daftar kata positif dan negatif untuk menentukan makna suatu teks.

2. Pendekatan Berbasis Pembelajaran Mesin.

Analisis sentimen berbasis machine learning adalah metode dalam pemrosesan bahasa alami (NLP) yang menggunakan algoritma pembelajaran mesin untuk mengklasifikasikan teks sebagai positif, negatif, atau netral. Tidak seperti pendekatan berbasis leksikon yang mengandalkan daftar kata dengan makna tertentu, metode berbasis machine learning mempelajari pola sentimen dari data yang telah diberi label (supervised learning) atau mengidentifikasi pola dalam data tanpa label (unsupervised learning). Beberapa algoritma yang digunakan seperti Naïve Bayes, SVM, Decision Tree dan Random Forest.

3. Pendekatan Berbasis Deep Learning.

Pendekatan ini menggunakan arsitektur jaringan saraf tiruan (neural networks). Contoh model yang sering digunakan antara lain Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), dan Transformers (BERT, GPT-3, RoBERTa) [5].

Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah bidang ilmu kecerdasan buatan yang berfokus pada pengolahan dan pemahaman bahasa alami komputer. NLP memiliki peran penting dalam menjembatani komunikasi antara manusia dan mesin, memungkinkan komputer untuk mengekstrak makna dari teks dan suara melalui pendekatan linguistik dan pembelajaran mesin [5]. Dengan perkembangan teknologi informasi yang sangat pesat saat ini, NLP telah menjadi komponen yang sangat penting dalam sistem digital, mulai dari pencarian informasi hingga interaksi berbasis teks melalui chatbot dan asisten virtual.

Teknologi informasi (TI) menyediakan landasan komputasi yang mendukung penerapan NLP dalam skala yang besar. Melalui infrastruktur TI, NLP dapat diintegrasikan dalam sistem informasi organisasi, analisis big data, hingga pengembangan aplikasi berbasis kecerdasan buatan. NLP mampu mengotomatisasi komunikasi serta menganalisa data teks dalam jumlah yang besar dan menjadikan solusi yang tepat di era transformasi digital. Dalam konteks ini, NLP tidak hanya dipandang sebagai alat teknis namun juga bagian dari strategi manajemen pengetahuan dan inovasi layanan digital [6]. Teori deep learning telah membawa revolusi dalam NLP modern dengan memperkenalkan arsitektur *neural network* seperti *Recurrent Neural Networks* (RNN), *Long Short Term Memory* (LSTM), dan *Transformer* yang memungkinkan pemodelan konteks secara lebih mendalam dan akurat [7]. Dalam perkembangan NLP, pendekatan berbasis Transformer seperti BERT (*Bidirectional Encoder Representations from Transformers*) telah meningkatkan kemampuan mesin dalam memahami konteks semantik dari teks. Hal ini ditunjukkan bagaimana representasi kontekstual yang dihasilkan BERT mampu meningkatkan akurasi dalam berbagai tugas NLP, seperti klasifikasi teks dan ekstraksi entitas [8]

NLP juga berperan penting dalam membentuk kerangka penelitian yang mencakup analisis kebutuhan pengguna, pemilihan pendekatan algoritma, serta perancangan sistem berbasis NLP. Analisis kebutuhan mencakup identifikasi jenis data teks yang relevan, seperti ulasan pelanggan, transkrip, atau pertanyaan pengguna. Pemilihan metode dapat disesuaikan antara pendekatan *deep learning* atau model klasik, tergantung kompleksitas dan tujuan penelitian. Desain sistem mencakup pemilihan arsitektur, *pipeline* pemrosesan teks, serta integrasi dengan sistem TI lainnya.

Penelitian ini secara umum terdiri dari variabel input berupa data teks mentah (misalnya komentar atau dokumen), proses seperti tokenisasi, vektorisasi, dan klasifikasi (dengan model seperti BERT atau SVM), serta output berupa informasi yang dimaknai (seperti sentimen, entitas, atau kategori). Penilaian terhadap sistem dilakukan melalui indikator seperti akurasi model, kecepatan respons, dan tingkat kepuasan pengguna.

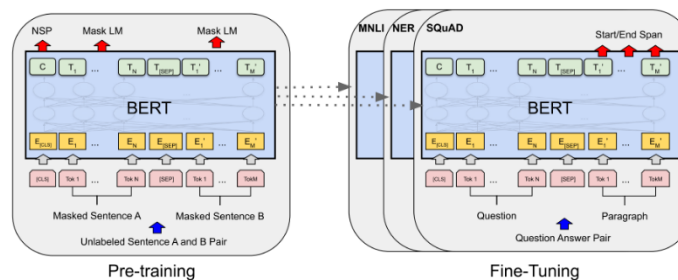
BERT (Bidirectional Encoder Representations from Transformers)

Bidirectional Encoder Representations from Transformers (BERT) merupakan teknik pembelajaran mesin yang dikembangkan oleh Google berbasis Transformer untuk pra-pelatihan pemrosesan bahasa alami (*Natural Language Processing*). BERT merupakan model representasi kata kontekstual yang dilatih sebelumnya berdasarkan MLM (*Masked Language Model*), menggunakan transformers dua arah. BERT adalah struktur *encoderdecoder* transformator dua arah

multi-layer. Transformer mengikuti keseluruhan arsitektur ini menggunakan stacked self-attention dan point-wise, terhubung sepenuhnya untuk *encoder* dan decoder. BERT membuat representasi kontekstual dari masukan teks dengan menggunakan *encoder*. Proses dimulai dengan tokenisasi teks, yang membentuk token kecil seperti kata, sub-kata, atau karakter. Selanjutnya, proses *embedding* digunakan untuk mengubah tiap token menjadi vektor kata. Proses ini menggunakan *embedding* yang telah dilatih sebelumnya oleh model. Kemudian, *encoder* BERT, yang terdiri dari berbagai layer transformer, bekerja pada token-token tersebut. Untuk membuat representasi kontekstual, setiap lapisan melakukan operasi multi-head self-attention dan full-connected feedforward networks.

BERT bisa dikatakan unik karena dapat memahami konteks secara bidireksional dan menangkap hubungan kontekstual yang lebih kuat dengan memperhatikan token kiri dan kanan. Proses ini berulang pada setiap layer *encoder* yang ditumpuk, yang menghasilkan pemahaman teks input yang semakin hierarkis [9]. Seiring berjalannya waktu, *Bidirectional Representation from Transformers* (BERT) yang merupakan model berbasis *Transformers* hadir dengan beberapa kelebihan utama, seperti kemampuan untuk melakukan parallel *training* yang berdampak pada efisiensi waktu untuk *training* model, serta kemampuan memahami konteks secara lebih baik secara dua arah [10].

Dalam praktiknya, BERT melakukan dua fase dalam prosesnya yaitu, *pre-training* untuk pemahaman bahasa dan *fine-tuning* untuk tugas tertentu. BERT dapat memahami bahasa dengan melatih mekanisme *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Dua fase yang digunakan di dalam BERT dapat dilihat pada gambar berikut :



Gambar 1. Arsitektur BERT [8]

Gambar 1. menunjukkan selama proses *pre-training*, model dilatih pada data yang tidak berlabel. Sedangkan selama *fine-tuning*, model BERT pertama kali diinisialisasi dengan parameter yang telah dilatih sebelumnya, dan semua parameter menggunakan data berlabel dari tugas *downstream*. Setiap tugas *downstream* memiliki model *fine-tuned* terpisah, meskipun diinisialisasi dengan parameter *pre-trained* yang sama.

IndoBERT

IndoBERT merupakan model bahasa berbasis *Transformer* yang dirancang untuk bahasa Indonesia. IndoBERT merupakan adaptasi dari BERT yang pada awalnya dikembangkan untuk bahasa Inggris. IndoBERT adalah *pre-trained* model yang dikembangkan dengan 2,4 juta langkah atau 180 epoch, sehingga menghasilkan kinerja yang kuat pada berbagai tugas NLP. Pre-trained model dilatih menggunakan *masked language modeling* (MLM) dan *next sentence prediction* (NSP). Model ini menggunakan arsitektur *transformer* dengan 12 lapisan dan 768 unit pemrosesan. IndoBERT dilatih dengan kosakata *WordPiece* bahasa Indonesia yang terdiri atas 31.923 token.

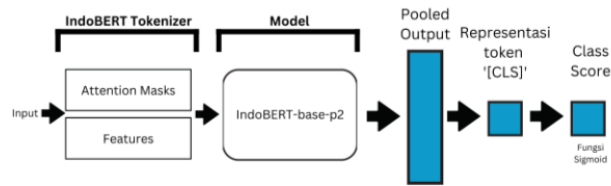
Setelah melalui proses tokenisasi, setiap token dari teks diubah menjadi representasi numerik atau *embedding* yang memungkinkan model untuk memahami konteks dan makna dari kata-kata dalam kalimat. Pada indoBERT, setiap token diubah menjadi vektor berdimensi tetap biasanya berukuran 768 dimensi yang mewakili hubungan semantik dan konteks kata tersebut dalam kalimat. *Embedding* ini mencakup informasi kontekstual dari setiap kata, memungkinkan algoritma untuk menangkap makna yang lebih mendalam dari teks. Selain itu, indoBERT juga menambahkan token khusus seperti [CLS] di awal dan [SEP] di akhir kalimat, yang masing-masing memberikan informasi tambahan untuk tugas klasifikasi dan pemisahan kalimat [11].

Beberapa varian IndoBERT yang telah dikembangkan termasuk IndoBERT-base, IndoBERT-large, dan IndoBERTlite. Varian dasar berisi 12 lapisan dan sekitar 125 juta parameter, sedangkan varian besar memiliki 24 lapisan dan sekitar 340 juta parameter [12].

IndoBERT – BC

IndoBERT *Base Classification* (IndoBERT-BC) ini diimplementasikan sesuai dengan ilustrasi pada Gambar 2 yang dimulai dengan memproses *input* menggunakan IndoBERT *tokenizer* untuk menjadi *features* berupa token *input* dan *attention masks*. *Features* dan *attention masks* kemudian diolah oleh model IndoBERT sehingga menghasilkan *pooled output*. Nilai representasi dari token khusus '[CLS]' pada indeks pertama *pooled output* kemudian diambil sebagai *class score*. *Class score* kemudian diolah dengan fungsi *sigmoid* untuk klasifikasi biner yang ditentukan oleh

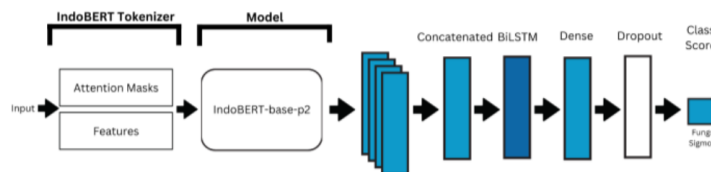
score_threshold. Alur *fine tuning approach* yang memanfaatkan *pooled output* akan menjadi *baseline* dari penelitian sebagai *IndoBERT Base Classification (IndoBERT-BC)* [10].



Gambar 2. Arsitektur Model IndoBERT-BC [10]

IndoBERT – BiC

Selain *fine tuning approach* dalam melatih IndoBERT, pendekatan lain yang digunakan adalah *feature-based approach* yang memanfaatkan *hidden states* tanpa mengubah parameter model. Pendekatan *feature-based* melalui penggabungan empat *hidden state* terakhir sebagai fitur dalam melakukan tugas klasifikasi telah ditunjukkan mendapatkan skor F1 tertinggi dibandingkan pendekatan lainnya. *Feature based approach* juga dilakukan dengan menggunakan empat *hidden state* terakhir yang berisi informasi kontekstual teks secara menyeluruh sebagai fitur. Fitur ini perlu diproses lebih lanjut terlebih dahulu untuk dapat menemukan konteks kalimat melalui hubungan token - token dalam urutan yang panjang. Salah satu model yang dapat digunakan adalah LSTM, karena model ini mampu memperhitungkan korelasi antara token. LSTM memperhitungkan bobot token untuk token selanjutnya sehingga model ini dapat menangkap konteks kalimat dari satu arah urutan (*forward features*). Namun, LSTM masih memiliki keterbatasan dalam mengatasi ketergantungan terhadap urutan yang membutuhkan pemahaman konteks baik dari token sebelumnya (*backward features*) dan token setelahnya (*forward features*). Untuk mendapatkan pemahaman konteks yang lebih kaya, BiLSTM yang mampu memahami konteks kalimat dari kedua arah urutan secara bersamaan digunakan. Seperti yang ditunjukkan pada Gambar 3 dapat dilihat perbedaan model IndoBERT *BiLSTM Classification* (IndoBERT-BiC) dengan model IndoBERT-BC [10].

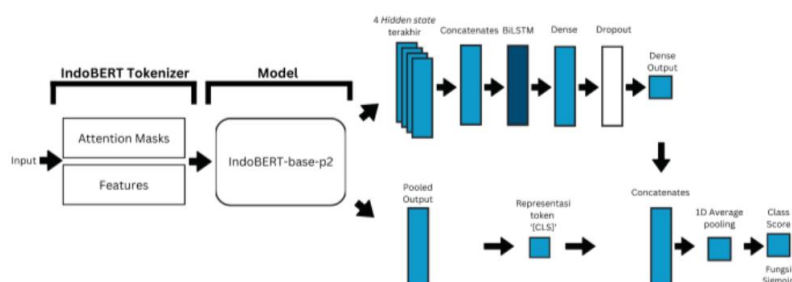


Gambar 3. Arsitektur Model IndoBERT-BiC [10]

IndoBERT – CC (Concatenation Classification)

Pendekatan *Concatenation Classification* merupakan salah satu metode dalam analisis sentimen yang menggabungkan berbagai representasi teks sebelum diklasifikasikan menggunakan model machine learning atau deep learning. Metode ini dapat meningkatkan akurasi prediksi dengan menggabungkan berbagai sumber informasi, seperti *embedding* dari IndoBERT dengan fitur tambahan dari teknik lain seperti TF-IDF atau *word embedding* lainnya.

Penggabungan *fine tuning approach* dengan *feature-based approach* memiliki potensi untuk meningkatkan performa model. Gambar 4 menunjukkan arsitektur model IndoBERT *Concatenation Classification* (IndoBERT-CC) yang merupakan penggabungan dua pendekatan yang dijelaskan sebelumnya. Pembentukan model ini diawali dengan melakukan *fine tuning* parameter pada model IndoBERT melalui pelatihan *dataset*. Langkah selanjutnya adalah menjalankan kedua proses yang dilakukan oleh model sebelumnya yakni mengekstraksi representasi token '[CLS]' pada *pooled output* dan melakukan *feature extraction* pada empat *hidden state* terakhir untuk diolah menggunakan BiLSTM. Berbeda dengan model IndoBERT-BC dan IndoBERT-BiC, terdapat - tahapan penggabungan kedua hasil proses tersebut yang kemudian diikuti dengan reduksi dimensi menggunakan 1 *Dimension average pooling* (1D *average pooling*). Hasil reduksi tersebut kemudian dihitung menggunakan fungsi *sigmoid* sehingga didapatkan *class score* [10].



Gambar 4. Arsitektur Model IndoBERT-CC [10]

Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi machine learning dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada *confusion matrix* yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative* [13].

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 5. *Confusion Matrix* [14]

Keterangan:

- a. *True Positive (TP)* : Prediksi positif dan itu benar.
- b. *True Negative (TN)* : Prediksi Negatif dan itu benar.
- c. *False Positive (FP)* : (Kesalahan Tipe 1) Prediksi positif dan itu salah.
- d. *False Negative (FN)* : (Kesalahan Tipe 2) Prediksi salah dan itu salah.

Dengan dasar *Confusion Matrix* ini kemudian dapat dilakukan perhitungan nilai *accuracy*, *presicion*, *F1-score* dan *recall*.

1. *Accuracy*

Accuracy menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \times 100 \dots\dots\dots(1)$$

2. *Precision*

Precision merupakan tingkat ketepatan model dalam memprediksi kelas tertentu (berapa banyak prediksi positif yang benar). *Precision* mengukur semua prediksi pada label tertentu berapa persen yang prediksinya benar. Atau dalam bahasa yang lebih teknis, *precision* adalah rasio antara prediksi positif yang benar (*true positive*) dengan total prediksi positif (baik yang benar maupun salah) pada setiap label.

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

3. *Recall*

Recall adalah tolak ukur seberapa baik model dalam mendeteksi semua label benar dari kelas tertentu. *Recall* mengukur seberapa banyak prediksi positif dari semua data yang aktualnya positif dari kelas tertentu. Atau secara teknis, adalah rasio antara prediksi positif yang benar dengan total jumlah data yang sebenarnya positif.

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

4. *F1-Score*

F1-Score adalah metrik yang menggabungkan *precision* dan *recall*. *F1-score* sangat berguna ketika kita menghadapi ketidakseimbangan kelas dalam dataset. Nilai *F1-score* adalah rata-rata antara *precision* dan *recall*. Dalam konteks sentimen , *F1-Score* memberikan Gambaran tentang seberapa baik model kita dalam mengklasifikasikan sentimen positif maupun negatif secara akurat.

$$F1 = 2 \times \frac{precision \times recall}{precision+ recall} \dots\dots\dots(4)$$

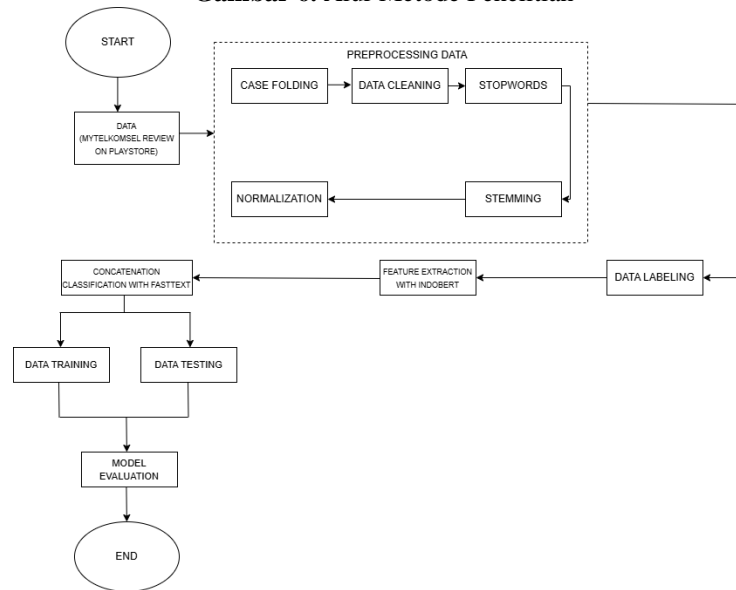
Selain menggunakan *confusion matrix* penelitian ini juga menggunakan nilai polaritas untuk melakukan pelabelan data ke dalam lima kategori, yaitu sangat positif, positif, netral, negatif, atau sangat negatif. Nilai polaritas ini biasanya dihitung dengan rumus :

$$Polaritas = \frac{\sum Skor \text{ Kata}}{\text{Jumlah Kata Bernilai Sentimen}} \dots\dots\dots(5)$$

3. Metode Penelitian

Penelitian ini menggunakan metode *text classification* berbasis *deep learning* memanfaatkan model IndoBERT dan teknik *concatenation classification*. Tujuannya adalah untuk mengklasifikasikan sentimen pengguna (sangat positif, positif, netral, negatif, dan sangat negatif) [20] terhadap aplikasi MyTelkomsel secara akurat dan otomatis. Proses pada penelitian ini dapat dilihat pada Gambar 6.

Gambar 6. Alur Metode Penelitian



Preprocessing Data

Pada tahapan ini, dilakukan *preprocessing* data untuk menghilangkan noise dan membersihkan data dari unsur-unsur yang tidak relevan seperti symbol, tanda baca, emoji, angka, serta mengkonversi huruf besar-kecil.

a. *Case Folding*

Tahapan ini adalah langkah untuk megubah semua karakter menjadi huruf kecil. Proses ini dilakukan karena data yang di dapatkan tidak selalu memiliki struktur yang teratur dan konsisten dalam penggunaan huruf kapital maupun non kapital. Oleh karena itu case folding diterapkan untuk mengkonversi teks menjadi huruf kecil [21].

Tabel 1. Contoh *Case Folding*

Text Sebelum <i>Case Folding</i>	Text Sesudah <i>Case Folding</i>
Selama Download MyTelkomsel jadi gak ribet sih beli paket data, tapi terkadang ada nge bug nya	selama download mytelkomsel jadi gak ribet sih beli paket data, tapi terkadang ada nge bug nya

b. *Data Cleaning*

Data Cleaning digunakan untuk menghapus atribut yang mempengaruhi kualitas analisis data dengan cara menghilangkan atribut tertentu seperti angka, symbol, URL, nama pengguna (@username), tagar (#), spasi berlebihan, tanda baca, emoji, dan karakter yang diulang dalam kalimat [22]. Proses ini menggunakan library PythonRegEx untuk mengidentifikasi karakter-karakter yang perlu dihapus.

Tabel 2. Contoh Data Cleaning

Text Sebelum <i>Data Cleaning</i>	Text Sesudah <i>Data Cleaning</i>
Aplikasi yang sangat membantu 👍👍 Loading doang, jalan kagak 🗨	Aplikasi yang sangat membantu Loading doang, jalan kagak

c. *Stopwords*

Stopwords membantu menghilangkan data yang tidak diperlukan. Tujuan dari stopword ini adalah untuk menghapus kata-kata yang sering muncul namun tidak memiliki makna yang signifikan, sehingga kata-kata yang tersisa dalam data hanyalah data yang akurat dan bermakna. Contoh nya seperti “dan”, “atau”, “tetapi”, “maka”, “karena”, “yang”, “adalah”, bagaimana”, “kenapa” [23].

Tabel 3. Contoh Stopwords

Text Sebelum Stopwords	Text Sesudah Stopwords
selama download mytelkomsel jadi gak ribet sih beli paket data, tapi terkadang ada nge bug nya	download mytelkomsel gak ribet beli paket data, terkadang bug

d. *Stemming*

Stemming merupakan proses menghilangkan imbuhan yang terdapat pada teks sehingga mengubahnya menjadi kata asli [21]. Tujuannya adalah untuk menghindari pembacaan kata berimbuhan yang berulang-ulang saat pemodelan yang dapat menyebabkan noise atau gangguan dalam analisis data. Dengan kata lain, stemming membantu menyederhanakan variasi kata sehingga model dapat bekerja lebih efisien dan akurat.

Tabel 4. Contoh *Stemming*

No	Slang	Formal
1	woww	wow
2	kaka	kakak
3	ka	kak
4	daah	dah
5	jgn	jangan
6	rb	ribu
7	bangett	banget
...
15007	gaharus	enggak harus

e. *Normalization*

Pada tahap ini merupakan langkah mengubah teks ke dalam bentuk yang lebih konsisten sesuai dengan kaidah yang berlaku. Dalam penelitian ini digunakan *Natural Language Toolkit* (NLTK) yang dipadukan dengan *Colloquial Indonesian Lexicon*. Sampel kata baku dapat dilihat dari tabel 5.

Tabel 5. Contoh *Normalization*

Text Sebelum Stemming	Text Sesudah Stemming
['download', 'mytelkomsel', 'gak', 'ribet', 'beli', 'paket', 'data', 'terkadang', 'bug']	download mytelkomsel gak ribet beli paket data, kadang bug

Data Labeling

Data ulasan yang telah dibersihkan selanjutnya diberi label sentimen. Setelah teks dinormalisasi text akan dihitung nilai polaritasnya yang menggambarkan seberapa positif, negatif, dan netral teks tersebut. Nilai polaritas akan didefinisikan sebagai berikut:

- Sangat Positif : $0.5 < x \leq 1$
- Positif : $0 < x \leq 0.5$
- Netral : $x = 0$
- Negatif : $-0.5 \leq x < 0$
- Sangat Negatif : $-1 \leq x < -0.51$

Tabel 6. Tabel Nilai Polaritas

Polaritas (Nilai)	Label
0.51 to 1.00	Sangat Positif
0.01 to 0.50	Positif
0.00	Netral
-0.50 to -0.01	Negatif
-1.00 to -0.51	Sangat Negatif

Contoh ulasan: “aplikasi lemot parah”

- aplikasi : 0 (netral)
- lemot : -0.7 (sangat negatif)

- parah : -0.9 (sangat negatif)

Skor kata di atas diperoleh dari kamus sentimen (*lexicon*), yaitu daftar kata yang sudah diberi skor oleh para ahli atau diperoleh dari data pelatihan. *Lexicon* yang digunakan dalam penelitian ini adalah *library vaderSentiment-id*. *VaderSentiment-id* adalah kamus Bahasa Indonesia dari *VADER*, yang dirancang untuk menganalisis sentimen informal, seperti: Bahasa gaul, singkatan, ulasan aplikasi, komentar media sosial dan tweet.

Tabel 7. Contoh Kamus *Lexicon*

Kata	Skor Kata
bagus	0.7
enak	0.6
hebat	0.8
puas	0.5
ramah	0.6
senang	0.6
buruk	-0.7
lambat	-0.6
mengecewakan	-0.8
mahal	-0.5
jelek	-0.7

Untuk menghitung nilai polaritas, dapat menggunakan persamaan (5) :

$$Polaritas = \frac{(-0.7) + (-0.9)}{2} = -0.8$$

Berdasarkan contoh perhitungan polaritas di atas maka untuk kalimat “aplikasi lemot parah” akan diberikan label “Sangat Negatif”

Feature Extraction with IndoBERT

Pada langkah ini teks akan diubah menjadi vektor numerik menggunakan model IndoBERT. Vektor tersebut merepresentasikan makna semantik dari teks, sehingga informasi dan hubungan antar kata dapat dipertahankan. Proses ini dilakukan melalui *embedding* yang dihasilkan oleh IndoBERT, yang secara otomatis menangkap pola, konteks, serta makna dalam bahasa Indonesia. Teks yang telah melalui *preprocessing* kemudian diproses menggunakan *tokenizer* dari IndoBERT. *Tokenizer* ini memecah teks menjadi token-token lebih kecil dengan menggunakan teknik *WordPiece*, yang memungkinkan representasi kata secara lebih fleksibel, terutama untuk kata-kata yang jarang muncul. Selain itu, token [CLS] secara otomatis ditambahkan di awal teks untuk mewakili keseluruhan isi teks, sedangkan token [SEP] ditambahkan di akhir untuk menandai batas akhir input. Proses ini menghasilkan urutan token yang siap digunakan oleh model IndoBERT untuk pemrosesan lebih lanjut.

Tabel 8. Contoh *Tokenization*

Text Sebelum <i>Tokenization</i>	Text Sesudah <i>Tokenization</i>
download mytelkomsel gak ribet beli paket data, kadang bug	[CLS], download, mytelkomsel, gak, ribet, beli, paket, data, kadang, bug,[SEP]

Token di atas lalu akan diubah menjadi angka unik disebut *Token ID* berdasarkan *vocabulary* IndoBERT.

Concatenation Classification

Pada tahap ini, ulasan yang sama juga diproses otomatis menggunakan *FastText*, yaitu metode *word embedding* yang lebih ringan dan cepat. *Library* bisa diambil dari <https://fasttext.cc/>. *FastText* menghasilkan representasi vektor yang lebih kecil (umumnya berdimensi 300). Setelah kedua vektor diperoleh vektor hasil IndoBERT dan vektor hasil *FastText* kedua representasi ini kemudian digabungkan (*concatenated*) secara horizontal, sehingga membentuk satu vektor baru yang panjangnya merupakan penjumlahan dari kedua dimensi awal (contohnya, 768 + 300 = 1068 dimensi).

Data Training

Data *training* digunakan untuk melatih model klasifikasi. Pembagian data dilakukan dengan dua skenario yaitu rasio 80% untuk *training* dan 20% untuk *testing* dan rasio 80% data *training*, 10% data *testing*, dan 10% data validasi. Model kemudian dilatih untuk mempelajari hubungan antara vektor input dengan label sentimen menggunakan fungsi *Logistic Regression*.

Data Testing

Data *testing* adalah sekumpulan data yang telah dipisahkan dari data *training* di awal, dengan rasio 20% dan 10% dari seluruh dataset. Setiap sampel dalam data *testing* telah diproses sehingga berbentuk vektor hasil *concatenation* antara *embedding* IndoBERT dan FastText.

Model Evaluation

Evaluasi model dilakukan dengan menggunakan sejumlah metrik, antara lain *accuracy*, *precision*, *recall*, dan *F1-score* berdasarkan persamaan (1), (2), (3), (4).

4. Hasil Penelitian

Pada bagian ini akan dijelaskan hasil dari penelitian yang telah dilakukan. Hasil dari penelitian ini terbagi menjadi beberapa tahapan sebagai berikut:

Dataset yang Digunakan

Tahapan ini menarik dataset ulasan aplikasi MyTelkomsel di PlayStore dari tahun 2022 sampai tahun 2024 menggunakan fungsi *google_play_scrapper*, dengan total 30.000 data dengan pembagian 10.000 data setiap tahunnya. Dataset dapat dilihat pada tabel berikut :

Tabel 9. Tabel Dataset

userName	ulasan	score	year
Putri Anna	sangat membantu dan memudahkan	5	2022
julius rovi	pdhal pake kartu sakti tapi paket nya mahal2,udh lama pake aplkasi telkomsel tapi makin lama makin mahal harga combo sakti nya.	1	2022
Mila Sari	Kenapa setengah setelah di update gak ada cek'in lagi.... Kesel kali gw Tiap hari buka my Telkomsel gak ada cek'in nya Dapat bonus apa apa aja gak ada lagi Pelit kali my Telkomsel sekarang ini	1	2022
Tria Puspita	sangat buruk jaringannya	5	2022
Mejoza	ok membantu	5	2022
Rizki Ramadhan	paket elit sinyal pas mati lampu sulit 📶 poor	1	2022
Anis Fuad HR.	bagus	5	2022
Masshodiqin 47	Kenapa ya Daily login nya ga aktif lagi dari awal bulan desember, pas di update terbaru ga bisa login daily lagi....apakah ada masalah? untuk sekarang bintang 1...kalau dah bener dibalikin bintang5, makasih.	1	2022
Kodir Jae	Pulsa hilang entah kemana...walaupun gk seberapa .	1	2022
Zuhriyani	Memudahkan segalanya, komunikas lancar, memudahkan utk menolong	5	2022
Ishak	tmn yg butuh (pulsa dan kuota), keperluan pembayaran penting, akses mudah, fiturnya layanannya byk, informasi byk, dan berlimpah promonya. 👍👍👍		
AKBAR Maulana	Untuk hadiah check-in kok di hilangin sih... 😞 Jdi gk guna apk nya	1	2022
rahman tiga	Njir poin 650 hilang tanpa kbar mening gati xl aj la gua	1	2022

Menampilkan Hasil *Pre-Processing Data*

Pada tahap ini dilakukan pre-processing data pada data ulasan aplikasi MyTelkomsel di PlayStore meliputi proses case folding, data cleaning, stopwords, stemming, dan normalization. Hasilnya dapat dilihat pada Gambar 7.

userName	ulasan	score	year	teks_preprocess	normalisasi
199	cakep	5	2022	cakep	cakep
# aa rasyidi3	mohon tingkatkan jaringan s...	1	2022	mohon tingkat jaringan seri...	mohon tingkat jaringan seri...
* Mas Azam *	Telkonsel mantap	5	2022	telkonsel mantap	telkonsel mantap
[Low Batt]	Tidak diperlukan. Terpaksa ...	1	2022	perlu paksa pakai	perlu paksa pakai
[Yoo Ruu] Firmanime	APAAN DAH SEKRANG TELOKSEL...	1	2022	apa dah sekrang teloksel a...	apa deh sekarang teloksel ...
_Muhamadsarif	Netral	3	2022	netral	netral
_RINYD	Jaringan Telkonsel gua kena...	3	2022	jaringan telkonsel saya lem...	jaringan telkonsel saya lem...
01.Abdillah Farel Leksono X...	Harga ga sesuai kualitas	1	2022	harga ga sesuai kualitas	harga enggak sesuai kualitas
056 AllFines	kok ga bisa bayar CODnya si...	1	2022	kok enggak bayar codnya sih...	kok enggak bayar codnya sih...
06. Muhammad Hyoga Pratama	daily check in hilang, harg...	1	2022	daily check in hilang harga...	daily check ini hilang harg...

Gambar 7. Dataset yang Telah Melalui *Pre-Processing*

Menampilkan Hasil *Labeling Data*

Pada tahap ini dilakukan proses pelabelan sentimen terhadap data ulasan pengguna aplikasi MyTelkonsel yang telah melalui tahapan *preprocessing*. Label sentimen yang digunakan terdiri dari lima kategori, yaitu sangat positif, positif, netral, negatif, dan sangat negatif. Setiap kata dalam ulasan diberi skor polaritas, kemudian nilai rata-rata polaritas dihitung untuk menentukan kategori sentimen. Ulasan dengan polaritas antara 0.51 hingga 1.00 diberi label sangat positif, 0.01 hingga 0.50 diberi label positif, nilai 0 dianggap netral, nilai antara -0.01 hingga -0.50 diberi label negatif, dan antara -0.51 hingga -1.00 diberi label sangat negatif.

userName	ulasan	score	year	teks_preprocess	normalisasi	polaritas	label
Raisya dan ihsan	the best	5	2022	the best	the best	0.9695	sangat positif
Afdal Lape	isi pls dong	5	2022	isi pls dong	isi pulsa dong	0.0494	positif
Ridwan Muhammad	Harga mahal kualitas 0	1	2023	harga mahal kualitas	harga mahal kualitas	-0.35	negatif
Lina Mahadewi	Dari kemarin sy mencoba mas...	1	2024	kemarin sy coba masuk akun ...	kemarin saya coba masuk aku...	0.076	positif
Bintara Siaga	Gue suka aplikasi mytelkoms...	5	2023	gue suka aplikasi mytelkoms...	gue suka aplikasi mytelkoms...	0.4051	positif
malik Abdul aziz	Selesai di update malah tam...	3	2024	selesai update malah tambah...	selesai update malah tambah...	-0.4767	negatif
Herdidi	Aplikasi ini terlalu berat,...	3	2024	aplikasi terlalu berat kura...	aplikasi terlalu berat kura...	-0.6208	sangat negatif
Susilawati	Sangat mengecewakan dalam s...	4	2022	sangat kecewa sinyal	sangat kecewa sinyal	-0.7191	sangat negatif
ahri Maulana	Rusak!!!! aplikasinya banya...	1	2023	rusak aplikasi banyak bug l...	rusak aplikasi banyak bug l...	-0.6718	sangat negatif
Maulidiyah Manik	telkonselkeren 🍕	5	2022	telkonsel keren	telkonsel keren	0.1806	positif

Gambar 8. Hasil *Data Labeling*

Proses Pemodelan *IndoBERT Concatenation Classification*

Tahap pemodelan merupakan inti dari proses analisis sentimen, di mana data yang telah diproses dan diberi label digunakan untuk melatih model *machine learning*. Pada penelitian ini, digunakan model *IndoBERT Concatenation Classification* dengan *FastText* untuk mengklasifikasikan ulasan pengguna ke dalam lima kategori sentimen, yaitu sangat positif, positif, netral, negatif, dan sangat negatif.

a. Pemisahan Data

Sebelum proses pelatihan dilakukan, data terlebih dahulu dibagi menjadi dua bagian, yaitu data latih (training data) dan data uji (testing data) dengan perbandingan 80:20. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengevaluasi kinerja model

b. Load Model

Pada tahap ini dilakukan pemuatan dua model *embedding*, yaitu *IndoBERT* dan *FastText*, yang digunakan untuk mengubah teks menjadi representasi vektor numerik. Model *IndoBERT* dimuat menggunakan pustaka *transformers* dengan memanggil *BertTokenizer* dan *BertModel* dari model pra-latih *indobenchmark/indobert-base-pl*.

Tokenizer berfungsi untuk memecah kalimat ulasan menjadi token sesuai format yang dikenali oleh model *BERT*, sementara *BertModel* menghasilkan *embedding* dari token-token tersebut. Setelah dimuat, model *BERT* diatur ke mode evaluasi dengan perintah *bert_model.eval()* agar model tidak melakukan proses pelatihan ulang. Selain itu, juga digunakan model Bahasa Indonesia, yang dimuat melalui *fasttext.load_model* ("cc.id.300.bin"). Model ini menghasilkan *embedding* berbasis kata menggunakan pendekatan subword, sehingga tetap mampu menangani kata-kata tidak baku atau kata baru. Kedua model *embedding* ini berperan penting dalam tahap ekstraksi fitur sebelum proses klasifikasi dilakukan.

c. *Embedding Function*

Setelah model *embedding* dimuat, tahap selanjutnya adalah mendefinisikan fungsi *embedding* yang digunakan untuk mengubah data teks menjadi representasi numerik. Dalam penelitian ini, digunakan dua pendekatan *embedding*, yaitu *FastText* dan *IndoBERT*, yang masing-masing diimplementasikan dalam fungsi *get_fasttext_vector()* dan *get_bert_cls_embedding()*.

Fungsi *get_fasttext_vector(text)* bertujuan untuk menghasilkan representasi vektor dari kalimat menggunakan model *FastText* Bahasa Indonesia. Fungsi ini memanggil *ft_model.get_sentence_vector(text)* untuk memperoleh vektor berdimensi 300 yang mewakili keseluruhan kalimat.

Sementara itu, fungsi *get_bert_cls_embedding(text)* menggunakan model *IndoBERT* untuk menghasilkan *embedding* kontekstual. Teks terlebih dahulu ditokenisasi menggunakan tokenizer *IndoBERT*, kemudian diproses oleh model *bert_model*. Output yang digunakan adalah representasi dari token spesial [CLS] (*Classification token*) yang berada pada posisi pertama dari *last_hidden_state*. Token ini secara konvensional digunakan dalam arsitektur *BERT* untuk mewakili keseluruhan kalimat. Fungsi ini dikemas dengan konteks *torch.no_grad()* untuk memastikan bahwa proses hanya bersifat *inference* dan tidak memengaruhi bobot model.

Penggunaan dua fungsi *embedding* ini memungkinkan analisis perbandingan antara pendekatan berbasis *word embedding* tradisional dan kontekstual. Vektor hasil dari masing-masing fungsi kemudian digunakan sebagai input dalam model klasifikasi sentimen pada tahap selanjutnya.

d. *Concatenation Classification*

Setelah *embedding* dari *FastText* dan *IndoBERT* berhasil dihasilkan, tahap selanjutnya adalah menggabungkan (*Concatenation*) kedua representasi vektor tersebut menjadi satu vektor fitur utuh. Sebelum itu, terlebih dahulu dilakukan pembersihan kolom teks dengan memastikan tidak ada nilai kosong (NaN) dan seluruh isi kolom bertipe string agar dapat diproses oleh *tokenizer* dan model *embedding*.

Proses penggabungan dilakukan melalui fungsi *get_combined_vector(text)*, yang memanggil dua fungsi *embedding* sebelumnya: *get_fasttext_vector()* untuk menghasilkan vektor berdimensi 300 dan *get_bert_cls_embedding()* untuk menghasilkan vektor berdimensi 768 dari token [CLS]. Kedua vektor tersebut kemudian digabungkan menggunakan fungsi *np.concatenate()* sehingga menghasilkan vektor akhir berdimensi 1068.

Selanjutnya, seluruh data teks dari dataset latih dan uji diolah menggunakan fungsi tersebut untuk membentuk *x_train* dan *x_test*, yaitu *array numpy* yang berisi vektor-vektor fitur gabungan. Label sentimen yang telah ditentukan sebelumnya dimasukkan ke dalam variabel *y_train* dan *y_test*. Hasil akhir dari tahap ini adalah data fitur dan label yang siap digunakan dalam proses pelatihan dan evaluasi model klasifikasi.

```
# === Step 4: Gabungkan Embedding ===
# Pastikan kolom teks bersih
train_df['teks'] = train_df['teks'].fillna('').astype(str)
test_df['teks'] = test_df['teks'].fillna('').astype(str)

def get_combined_vector(text):
    ft_vec = get_fasttext_vector(text)
    bert_vec = get_bert_cls_embedding(text)
    return np.concatenate([bert_vec, ft_vec]) # concatenation: 768 + 300 = 1068

print("Extracting combined embeddings...")
X_train = np.array([get_combined_vector(t) for t in train_df['teks']])
y_train = train_df['label_id'].values

X_test = np.array([get_combined_vector(t) for t in test_df['teks']])
y_test = test_df['label_id'].values
```

Gambar 9. Proses *Concatenation*

e. *Pelatihan dan Evaluasi*

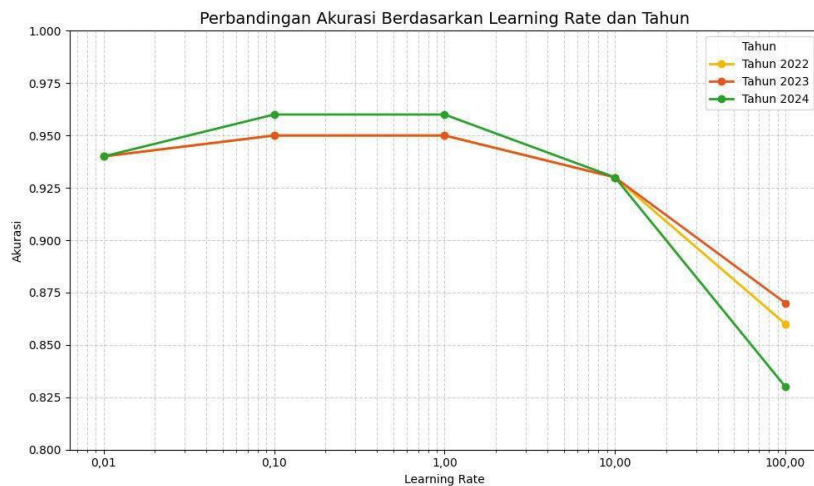
Pada proses ini dilakukan pelatihan dan evaluasi menggunakan *confussion matrix*. Tahap pelatihan menggunakan fungsi *logistic regression*. Model kemudian dilatih menggunakan data latih *x_train* dan label sentimen *y_train*, sehingga model dapat mempelajari pola hubungan antara data ulasan dan kelas sentimennya. Setelah proses pelatihan selesai, dilakukan tahap evaluasi untuk menguji performa model terhadap data uji *x_test*. Model yang telah dilatih digunakan untuk memprediksi sentimen, dan hasilnya disimpan dalam variabel *y_pred*. Selanjutnya, digunakan fungsi *classification_report* dari pustaka *sklearn.metrics* untuk membandingkan hasil prediksi dengan label sebenarnya (*y_test*) dan menampilkan metrik evaluasi seperti *precision*, *recall*, *f1-score*, serta jumlah data pada setiap kelas (*support*). Hasil evaluasi ini memberikan gambaran mengenai kemampuan model dalam mengklasifikasikan sentimen secara akurat dan menjadi dasar dalam menilai keefektifan model yang digunakan dalam penelitian ini.

Pengujian dilakukan menggunakan dua skenario. Skenario pertama menggunakan 80% data untuk pelatihan (*training*) dan 20% untuk pengujian (*testing*). Skenario kedua menggunakan 80% data untuk pelatihan (*training*), 10% untuk validasi, dan 10% (*testing*) untuk pengujian. Pada skenario pertama, eksperimen dilakukan terhadap data tahun 2022 hingga 2024 dengan berbagai konfigurasi *Max Iteration (Max_Iter)* dan *Learning Rate*. Hasil pengujian

menunjukkan bahwa performa terbaik dicapai pada tahun 2024, dengan kombinasi *Max_Iter* = 1000 dan *Learning Rate* = 1. Pada konfigurasi tersebut, model klasifikasi sentimen berhasil mencapai akurasi sebesar 0,96. Hal ini menunjukkan bahwa model mampu mengenali sentimen secara konsisten dan akurat pada data tahun tersebut. Jika dibandingkan dengan tahun-tahun sebelumnya, performa model pada tahun 2022 cenderung lebih rendah. Misalnya, pada konfigurasi dengan *Learning Rate* = 100, akurasi menurun tajam hingga 0,86, dan *f1-score* untuk kelas “Sangat Negatif” serta “Sangat Positif” juga menurun signifikan. Ini mengindikasikan bahwa nilai *learning rate* yang terlalu besar menyebabkan ketidakstabilan selama pelatihan, sehingga model gagal mempelajari representasi data secara optimal. Performa model pada tahun 2023 menunjukkan peningkatan yang cukup signifikan dibanding tahun 2022. Dengan kombinasi *Max_Iter* = 1000 dan *Learning Rate* = 1, akurasi model mencapai 0,95 dan *f1-score* tinggi di hampir semua kelas. Hal ini mengindikasikan bahwa model telah mulai menunjukkan stabilitas, terutama saat menggunakan nilai *learning rate* yang sesuai. Hasil pengujian pertama dapat dilihat pada tabel 10.

Tabel 10. Tabel Hasil Pengujian 1

Max_Iter	Learning Rate	Year	Accuracy	Eksperimen 80% Data Training dan 20% Data Testing														
				Very Negative			Negative			Neutral			Positive			Very Positive		
				Presisi	Recall	F1-Score	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1000	1	2022	0.95	0.94	0.92	0.93	0.93	0.93	0.93	0.94	0.97	0.96	0.97	0.96	0.96	0.92	0.92	0.92
1000	0.1	2022	0.95	0.94	0.91	0.92	0.92	0.93	0.93	0.95	0.96	0.96	0.97	0.97	0.97	0.93	0.91	0.92
1000	0.01	2022	0.94	0.94	0.91	0.92	0.91	0.92	0.92	0.95	0.96	0.95	0.96	0.96	0.92	0.9	0.91	
1000	10	2022	0.93	0.9	0.9	0.9	0.9	0.92	0.91	0.93	0.95	0.94	0.95	0.93	0.94	0.91	0.88	0.89
1000	100	2022	0.86	0.81	0.83	0.82	0.8	0.83	0.81	0.88	0.93	0.9	0.9	0.87	0.89	0.86	0.74	0.79
1000	1	2023	0.95	0.95	0.95	0.95	0.92	0.94	0.93	0.96	0.94	0.95	0.95	0.97	0.96	0.94	0.69	0.79
1000	0.1	2023	0.95	0.95	0.94	0.95	0.93	0.93	0.93	0.96	0.95	0.95	0.96	0.97	0.96	0.79	0.73	0.76
1000	0.01	2023	0.94	0.95	0.94	0.94	0.91	0.91	0.91	0.95	0.94	0.94	0.95	0.97	0.96	0.8	0.73	0.77
1000	10	2023	0.93	0.92	0.91	0.92	0.87	0.92	0.89	0.94	0.93	0.94	0.95	0.95	0.95	0.82	0.62	0.71
1000	100	2023	0.87	0.83	0.89	0.86	0.81	0.81	0.81	0.91	0.88	0.9	0.91	0.92	0.91	0.89	0.36	0.51
1000	1	2024	0.96	0.92	0.92	0.92	0.95	0.95	0.95	0.97	0.98	0.97	0.95	0.95	0.95	0.89	0.92	0.92
1000	0.1	2024	0.96	0.93	0.92	0.93	0.96	0.95	0.95	0.97	0.98	0.97	0.94	0.96	0.95	0.88	0.91	0.91
1000	0.01	2024	0.94	0.89	0.88	0.88	0.94	0.93	0.94	0.97	0.97	0.97	0.9	0.94	0.92	0.94	0.83	0.88
1000	10	2024	0.93	0.88	0.85	0.86	0.91	0.9	0.9	0.95	0.97	0.96	0.91	0.92	0.92	0.96	0.81	0.88
1000	100	2024	0.84	0.74	0.72	0.73	0.79	0.74	0.76	0.9	0.95	0.92	0.81	0.81	0.81	0.85	0.51	0.64



Gambar 10. Grafik Perbandingan Akurasi Berdasarkan *Learning Rate*

Berdasarkan grafik pada Gambar 10, perbandingan akurasi terlihat bahwa penggunaan nilai *learning rate* yang terlalu kecil seperti 0.01 menghasilkan performa yang kurang optimal dibandingkan nilai menengah seperti 0.1 atau 1. Walaupun masih tergolong stabil, model dengan *learning rate* 0.01 menunjukkan sedikit penurunan dalam akurasi dan *f1-score* di semua tahun pengujian. Penggunaan *learning rate* yang besar seperti 10 dan 100 cenderung menghasilkan performa yang lebih buruk. Pada nilai *learning rate* = 10, model mulai menunjukkan gejala ketidakstabilan. Misalnya, pada tahun 2022, 2023, dan 2024 akurasi hanya mencapai 0,93. Hal ini menunjukkan bahwa meskipun model masih bias belajar, proses pelatihannya terganggu oleh kecepatan pembaruan bobot yang terlalu besar.

Akurasi semakin mengalami penurunan saat *learning rate* ditingkatkan menjadi 100. Dapat dilihat pada tahun 2024 akurasi model turun drastis dengan akurasi yang hanya 0,84 dan *F1-score* pada kelas “Sangat Negatif” serta “Sangat Positif” masing-masing turun menjadi 0,73 dan 0,64. Hal ini menunjukkan bahwa nilai *learning rate* sebesar 100 terlalu

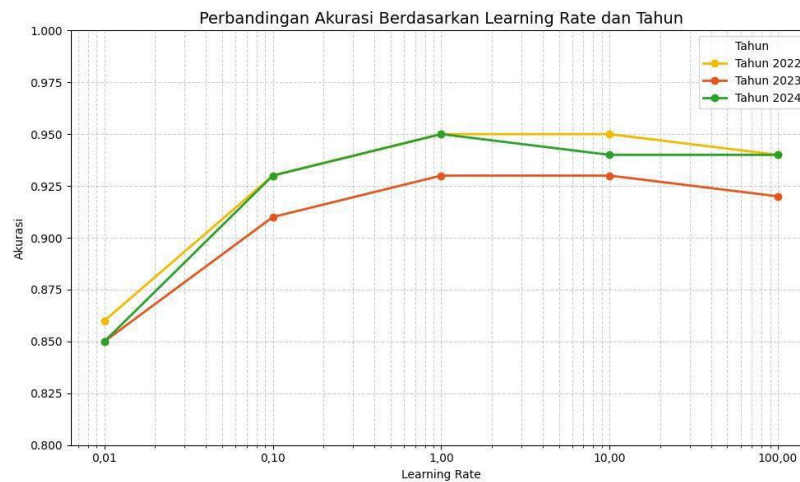
agresif, menyebabkan model kesulitan mencapai titik minimum loss secara stabil dan berdampak buruk pada kualitas prediksi.

Kombinasi *Max Iter* = 1000 dan *Learning Rate* = 1 memberikan performa terbaik dan paling konsisten selama tiga tahun pengujian, khususnya pada tahun 2024 yaitu mencapai 0,96. Hasil ini membuktikan bahwa pengaturan parameter pelatihan yang tepat memiliki peran yang sangat penting dalam meningkatkan kualitas dan kestabilan model klasifikasi sentimen. Dengan memilih parameter yang sesuai, model tidak hanya lebih akurat tetapi juga mampu mengklasifikasikan berbagai tingkat polaritas sentimen dengan lebih seimbang.

Selanjutnya dilakukan pengujian yang kedua dengan pembagian 80% data *training*, 10% data *validasi* dan 10% data *testing* menunjukkan hasil eksperimen klasifikasi sentiment menggunakan kombinasi parameter *max iteration* sebesar 1000 dan kombinasi nilai *learning rate* pada data dari tahun 2022 hingga 2024. Secara umum, akurasi model cukup tinggi pada seluruh tahun, berkisar antara 0.85 hingga 0.95. Nilai akurasi tertinggi (0.95) dicapai pada beberapa kombinasi, terutama dengan learning rate 1 dan 10, baik di tahun 2022 maupun 2024. Sebaliknya, akurasi terendah (0.85) muncul saat learning rate diatur ke 0.01 pada tahun 2023 dan 2024. Hasil pengujian kedua dapat dilihat pada tabel 11.

Tabel 11. Tabel Hasil Pengujian 2

Eksperimen 80% Data Training 10% Validasi dan Data Testing 10%																		
Max. Iter	Learning Rate	Year	Accuracy	Very Negative			Negative			Neutral			Positive			Very Positive		
				Precisi	Recall	F1-Score	Precisi	Recall	F1-Score	Precisi	Recall	F1-Score	Precisi	Recall	F1-Score	Precisi	Recall	F1-Score
1000	1	2022	0.95	0.91	0.95	0.93	0.94	0.94	0.94	0.96	0.97	0.97	0.98	0.96	0.97	0.91	0.95	0.93
1000	0.1	2022	0.93	0.86	0.94	0.9	0.93	0.9	0.91	0.96	0.97	0.97	0.93	0.94	0.94	0.93	0.91	0.92
1000	0.01	2022	0.86	0.78	0.87	0.82	0.82	0.8	0.81	0.89	0.89	0.89	0.89	0.89	0.89	0.83	0.74	0.78
1000	10	2022	0.95	0.91	0.94	0.92	0.93	0.94	0.94	0.96	0.97	0.97	0.97	0.96	0.97	0.96	0.89	0.92
1000	100	2022	0.94	0.9	0.95	0.93	0.93	0.9	0.91	0.95	0.97	0.96	0.98	0.97	0.98	0.92	0.88	0.9
1000	1	2023	0.93	0.92	0.92	0.92	0.88	0.93	0.9	0.94	0.91	0.92	0.96	0.95	0.96	0.92	0.92	0.92
1000	0.1	2023	0.91	0.88	0.88	0.88	0.86	0.89	0.87	0.89	0.9	0.9	0.95	0.93	0.94	0.86	0.82	0.84
1000	0.01	2023	0.85	0.8	0.89	0.84	0.81	0.74	0.77	0.86	0.86	0.86	0.89	0.89	0.89	0.8	0.55	0.65
1000	10	2023	0.93	0.91	0.92	0.91	0.89	0.94	0.91	0.94	0.91	0.93	0.96	0.95	0.95	0.78	0.82	0.8
1000	100	2023	0.92	0.9	0.91	0.9	0.88	0.92	0.9	0.93	0.93	0.93	0.96	0.93	0.95	0.78	0.82	0.8
1000	1	2024	0.95	0.88	0.92	0.9	0.95	0.88	0.91	0.97	0.98	0.97	0.93	0.95	0.94	0.98	0.91	0.94
1000	0.1	2024	0.93	0.85	0.93	0.89	0.93	0.83	0.87	0.94	0.97	0.96	0.91	0.91	0.91	0.97	0.87	0.92
1000	0.01	2024	0.85	0.76	0.75	0.75	0.82	0.67	0.74	0.89	0.96	0.92	0.81	0.84	0.82	0.9	0.58	0.7
1000	10	2024	0.94	0.9	0.94	0.92	0.94	0.88	0.91	0.97	0.97	0.97	0.93	0.95	0.94	0.95	0.91	0.93
1000	100	2024	0.94	0.88	0.93	0.91	0.95	0.87	0.9	0.96	0.97	0.97	0.93	0.94	0.94	0.93	0.91	0.92



Gambar 11. Grafik Perbandingan Akurasi Berdasarkan *Learning Rate* Pengujian 2

Dapat dilihat pada Gambar 11. bahwa grafik menunjukkan pengaruh variasi *learning rate* terhadap akurasi model klasifikasi sentimen untuk data tahun 2022, 2023, dan 2024. Secara umum, akurasi model mengalami peningkatan seiring kenaikan nilai *learning rate* dari 0,01 hingga 1,0. Pada titik *learning rate* = 1,0, seluruh model mencapai performa terbaik, dengan akurasi tertinggi dicapai pada tahun 2024 sebesar 0.95. Hal ini menunjukkan bahwa pada nilai tersebut, model mampu belajar secara efisien dan mengonvergensi ke solusi optimal.

Peningkatan nilai *learning rate* lebih lanjut ke 10 dan 100 tidak menghasilkan peningkatan akurasi. Justru, pada tahun 2023 terlihat adanya penurunan akurasi yang menunjukkan bahwa nilai *learning rate* yang terlalu besar dapat menyebabkan proses pelatihan menjadi tidak stabil.

Pemilihan *learning rate* yang tepat menjadi sangat krusial dalam proses pelatihan model. Berdasarkan hasil eksperimen, *learning rate* sebesar 1,0 merupakan titik optimal yang memberikan keseimbangan antara kecepatan

pelatihan dan stabilitas konvergensi. Nilai ini terbukti menghasilkan akurasi tinggi dan konsisten pada berbagai data dari tahun yang berbeda, yang menunjukkan bahwa konfigurasi tersebut mampu menangkap pola sentiment secara efektif dalam beragam kondisi data.

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan terhadap analisis sentimen ulasan pengguna aplikasi MyTelkomsel menggunakan pendekatan gabungan *embedding IndoBERT* dan *FastText*, dapat disimpulkan bahwa:

1. Berdasarkan hasil eksperimen yang dilakukan terhadap data tahun 2022 hingga 2024, dapat disimpulkan: bahwa model ini mampu memberikan performa klasifikasi yang baik, dengan akurasi tertinggi mencapai 96% pada data tahun 2024 menggunakan kombinasi parameter *Max Iteration* sebesar 1000 dan *Learning Rate* sebesar 1.0. Ini menunjukkan bahwa penggabungan model *IndoBERT* dan *FastText* berhasil mengklasifikasikan ulasan ke dalam lima kategori sentimen (sangat negatif, negatif, netral, positif, sangat positif) secara akurat dan seimbang.
2. Model terbukti mampu menangani karakteristik bahasa yang tidak terstruktur, termasuk penggunaan bahasa informal dan variasi ekspresi dalam ulasan. Proses concatenation antara *embedding* dari *IndoBERT* dan *FastText* berhasil memperkaya representasi fitur teks, sehingga meningkatkan pemahaman konteks oleh model.
3. Uji coba dua skenario pembagian data (80:20 dan 80:10:10) menunjukkan bahwa nilai *learning rate* yang terlalu tinggi dapat menurunkan akurasi, sementara nilai *learning rate* yang tepat (yakni 1.0) secara konsisten memberikan hasil terbaik di seluruh tahun pengujian.

Daftar Pustaka

- [1] O. A. Irmawan, I. Budi, A. B. Santoso, and P. K. Putra, "Improving Sentiment Analysis and Topic Extraction in Indonesian Travel App Reviews Through BERT Fine-Tuning," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 13, no. 2, pp. 359–370, Jul. 2024, doi: 10.23887/janapati.v13i2.77028.
- [2] C. Fiarni and christell Cellose, "Sentiment Analysis Of Indonesian Video Streaming Application Services Reviews Using Fine-Tuning Indobert And Aspect Modeling," *Journal of Computer Sciences and Informatics*, no. 0, p. 1, 2024, doi: 10.5455/JCSI.20240915104407.
- [3] A. Naufal, "Analisis Ulasan Pengguna Aplikasi Kesehatan Digital dengan Aspect-Based Sentiment Analysis sebagai Strategi Peningkatan Kualitas Layanan menggunakan Machine Learning," 2025.
- [4] Bing Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press. . 2020. Accessed: Apr. 03, 2025. [Online]. Available: https://www.google.co.id/books/edition/Sentiment_Analysis/PdX7DwAAQBAJ?hl=en&gbpv=1&pg=PA12&printsec=frontcover&dq=posit
- [5] D. Jurafsky and J. H. Martin, "Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models Third Edition draft Summary of Contents."
- [6] A. Bharathi, "Natural Language Processing for Enterprise Applications," *Ushus-Journal of Business Management*, vol. 2022, no. 4, pp. 29–39, doi: 10.12725/ujbm.61.2.
- [7] A. Vaswani *et al.*, "Machine Learning," Jun. 2017, [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [8] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." [Online]. Available: <https://github.com/tensorflow/tensor2tensor>
- [9] A. Dwiyono, M. Fachrurrozi, J. Palembang-Prabumulih, K. Ogan Ilir, and S. Selatan, "Analisis Perbandingan Klasifikasi Intent Chatbot Menggunakan Deep Learning BERT, RoBERTa, dan IndoBERT," *Journal of Information System Research*, vol. 6, no. 1, pp. 605–616, 2024, doi: 10.47065/josh.v6i1.6051.
- [10] L. Bodhi Wijaya, Y. Nuraga Wicaksana, S. S. Widhiasari, and A. Saptawijaya, "Wijaya dkk: Pengembangan Model Deteksi Hoaks Berbahasa Indonesia Menggunakan Kombinasi IndoBERT dan BiLSTM Pengembangan Model Deteksi Hoaks Berbahasa Indonesia Menggunakan Kombinasi IndoBERT dan BiLSTM," 2024.
- [11] J. Pranata, S. Agustian, and E. Haerani, "Penggunaan Model Bahasa indoBERT pada Metode Random Forest Untuk Klasifikasi Sentimen Dengan Dataset Terbatas," *Technology and Science (BITS)*, vol. 6, no. 3, pp. 1668–1676, 2024, doi: 10.47065/bits.v6i3.6335.
- [12] Anugerah Simanjuntak *et al.*, "Research and Analysis of IndoBERT Hyperparameter Tuning in Fake News Detection," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 13, no. 1, pp. 60–67, Feb. 2024, doi: 10.22146/jnteti.v13i1.8532.
- [13] S. Kom. , M. K. Dr. Maria Susan Anggreany, "Confusion Matrix." Accessed: Apr. 09, 2025. [Online]. Available: <https://socs.binus.ac.id/2020/11/01/confusion-matrix/>
- [14] Kuncahyo Setyo Nugroho, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning." Accessed: Apr. 15, 2025. [Online]. Available: <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-supervised-machine-learning-bc4b1ae9ae3f>
- [15] S. Aras, M. Yusuf, R. Y. Ruimassa, E. A. B. Wambrauw, and E. B. Pala'langan, "Sentiment Analysis on Shopee Product Reviews Using IndoBERT," *Journal of Information Systems and Informatics*, vol. 6, no. 3, pp. 1616–1627, Sep. 2024, doi: 10.51519/journalisi.v6i3.814.

- [16] H. Imaduddin, F. Yusfida A'la, and Y. S. Nugroho, "Sentiment Analysis in Indonesian Healthcare Applications using IndoBERT Approach." [Online]. Available: www.ijacsa.thesai.org
- [17] M. Daffa, A. Fahreza, A. Luthfiarta, M. Rafid, M. Indrawan, and A. Nugraha, "Analisis Sentimen: Pengaruh Jam Kerja Terhadap Kesehatan Mental Generasi Z," *JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)*, vol. 5, no. 1, pp. 2723–1453, 2024, doi: 10.52158/jacost.715.
- [18] M. Isnain, G. N. Elwirehardja, and B. Pardamean, "Sentiment Analysis for TikTok Review Using VADER Sentiment and SVM Model," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 168–175. doi: 10.1016/j.procs.2023.10.514.
- [19] D. Chi, T. Huang, Z. Jia, and S. Zhang, "Research on sentiment analysis of hotel review text based on BERT-TCN-BiLSTM-attention model," *Array*, vol. 25, Mar. 2025, doi: 10.1016/j.array.2025.100378.
- [20] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "Socio-Affective Computing 5 A Practical Guide to Sentiment Analysis." [Online]. Available: <http://www.springer.com/series/13199>
- [21] H. K. Putra, M. Arif Bijaksana, and A. Romadhony, "Deteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia Menggunakan Metode IndoBERT."
- [22] O. I. Gifari, M. Adha, I. Rifky Hendrawan, F. Freddy, and S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *JIFOTECH (JOURNAL OF INFORMATION TECHNOLOGY)*, vol. 2, no. 1, 2022.
- [23] S. Aras, M. Yusuf, R. Y. Ruimassa, E. A. B. Wambrauw, and E. B. Pala'langan, "Sentiment Analysis on Shopee Product Reviews Using IndoBERT," *Journal of Information Systems and Informatics*, vol. 6, no. 3, pp. 1616–1627, Sep. 2024, doi: 10.51519/journalisi.v6i3.814.
- [24] A. Z. M. Noor, R. Gernowo, and O. D. Nurhayati, "Data Augmentation for Hoax Detection through the Method of Convolutional Neural Network in Indonesian News," *Jurnal Penelitian Pendidikan IPA*, vol. 9, no. 7, pp. 5078–5084, Jul. 2023, doi: 10.29303/jppipa.v9i7.4214.