

Sentiment Analysis Using LSTM Algorithm Regarding Grab Application Services in Indonesia

Akbar Rizky Gunawan ^{1*}, Rifda Faticha Alfa Aziza ^{2*}

* Informatika, Universitas Amikom Yogyakarta

akbarzeronf@students.amikom.ac.id¹, rifda@amikom.ac.id²

Article Info

Article history:

Received 2024-10-15

Revised 2025-02-18

Accepted 2025-02-19

Keyword:

Sentiment Analysis,
User Reviews,
Long Short-Term Memory,
Bi-Directional LSTM,
Multi-Head Attention.

ABSTRACT

This study aims to analyze the sentiment of user reviews for the Grab Indonesia application using Long Short-Term Memory (LSTM) algorithms. Two variants of LSTM, namely Stacked LSTM and Bi-Directional LSTM, were compared to determine the most effective model in classifying user review sentiments. Both models were enhanced with Multi-Head Attention mechanisms to capture more complex contextual relationships in sequential data. The data used consists of 2,000 user reviews collected through scraping from the Google Play Store, with sentiment labels of positive and negative. Data preprocessing included labeling, case folding, stopword removal, tokenization, stemming, and the application of the SMOTE technique to address class imbalance. The results show that the Bi-Directional LSTM model achieved the highest validation accuracy of 87%, with an F1-score of 0.90 for the negative class and 0.82 for the positive class, while the Stacked LSTM recorded an accuracy of 84%, with an F1-score of 0.87 for the negative class and 0.78 for the positive class. Overall, the Bi-Directional LSTM demonstrated better performance in identifying both negative and positive sentiments, providing a good balance between precision and recall. This study proves that Bi-Directional LSTM with Multi-Head Attention can improve sentiment analysis performance on user reviews of digital applications, with potential applications in various other platforms.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Sejak masuknya era digital, telah terjadi peningkatan signifikan dalam jumlah pengguna internet di seluruh dunia. Peningkatan ini sejalan dengan pertumbuhan jumlah data yang tersedia di internet. Ketersediaan data yang melimpah, terutama data teks, telah mendorong banyak penelitian dalam bidang *text mining* dan *Natural Language Processing (NLP)*. Salah satu bidang penelitian *NLP* yang populer adalah analisis sentimen. Analisis sentimen adalah proses komputasi untuk mengklasifikasikan data teks sesuai dengan sentimen yang terkandung di dalamnya. Popularitas penelitian analisis sentimen juga didorong oleh kebutuhan untuk memahami opini publik tentang topik tertentu. Proses analisis sentimen dipengaruhi oleh kualitas dataset yang digunakan. Untuk dataset yang terdiri dari kalimat-kalimat panjang, penanganannya memerlukan pendekatan yang berbeda dan

lebih kompleks [1]. Aplikasi seluler telah menjadi tulang punggung bagi berbagai layanan dan aktivitas dalam era digital. Salah satu contohnya adalah aplikasi *Grab*, yang telah menjadi salah satu layanan berbasis teknologi utama di sektor transportasi dan pengiriman makanan. Orang sering kali mengungkapkan opini dan ulasan mereka tentang layanan aplikasi seperti *Grab* melalui berbagai media, termasuk *Google Play Reviews*. Data ulasan ini dapat digunakan sebagai sumber utama dalam penelitian analisis sentimen [2].

Ojek online telah menjadi pilihan populer karena kemudahannya dalam mengakses layanan transportasi melalui aplikasi tanpa perlu datang ke tempat konvensional untuk memesan. Orang sering kali mengungkapkan opini dan ulasan mereka tentang layanan ojek *online* melalui berbagai media, termasuk *Google Play Reviews*. Melalui platform ini, pengguna dapat memberikan ulasan tentang kelebihan dan kekurangan layanan ojek *online*. Ulasan pengguna aplikasi

ojek *online* dapat bervariasi, mulai dari yang sangat positif hingga sangat negatif, tergantung pada pengalaman masing-masing pengguna. Selain memberikan ulasan, pengguna juga dapat memberikan rating berupa skor bintang dan menyatakan apakah mereka menyukai aplikasi tersebut atau tidak [3].

Analisis sentimen adalah salah satu solusi yang dapat digunakan untuk menentukan tingkat kepuasan pengguna dari data yang tidak terstruktur. Dalam konteks ini, *Deep Learning* muncul sebagai metode yang efektif untuk mengimplementasikan analisis sentimen. *Deep Learning* bekerja dengan mengekstraksi data menggunakan jaringan syaraf tiruan, di mana model belajar dengan menyesuaikan nilai error. Salah satu pendekatan *Deep Learning* yang umum digunakan untuk klasifikasi teks adalah *Long Short-Term Memory (LSTM)*. Dengan kemampuannya memahami konteks dan hubungan antar kata dalam teks, menjadi metode yang ideal untuk menganalisis sentimen ulasan pelanggan terhadap layanan Grab [4]. Variasi seperti *Stacked LSTM* dan *Bidirectional LSTM* juga menunjukkan potensi untuk meningkatkan performa dalam klasifikasi teks.

Pada penelitian sebelumnya, didapatkan hasil evaluasi bahwa algoritma *BiLSTM* memiliki performa yang lebih baik dibandingkan dengan *LSTM* pada data pengujian, dengan akurasi sebesar 78% untuk *BiLSTM* yang lebih tinggi dibandingkan 76% pada *LSTM*. Evaluasi dan perhitungan metrik menggunakan persamaan terkait menghasilkan nilai *precision* sebesar 0.69, *recall* sebesar 0.82, serta *F1-score* sebesar 0.75. Hasil ini mengindikasikan bahwa algoritma *BiLSTM* lebih unggul dalam mengidentifikasi sentimen dibandingkan algoritma *LSTM*, akan tetapi hasil yang didapatkan belum cukup baik terutama dalam keseimbangan antara presisi dan sensitivitas model [4].

Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap ulasan pengguna aplikasi *Grab* di platform *Google Play*, dengan fokus utama pada perbandingan performa dari model *Long Short-Term Memory (LSTM)*, penelitian ini akan membandingkan *Stacked LSTM* dan *Bidirectional LSTM* yang akan di tambahkan dengan teknik *Multi Head Attention* untuk menentukan hasil terbaik dalam mengidentifikasi sentimen pengguna, baik yang bersifat positif maupun negatif. Selain itu, penelitian ini juga akan mengevaluasi performa masing-masing model berdasarkan akurasi, *loss*, *precision*, *recall*, dan *f1-score*, untuk memberikan gambaran tentang seberapa efektif model *LSTM* dalam menangkap sentimen pengguna terhadap layanan yang diberikan oleh pengguna pada aplikasi *Grab*.

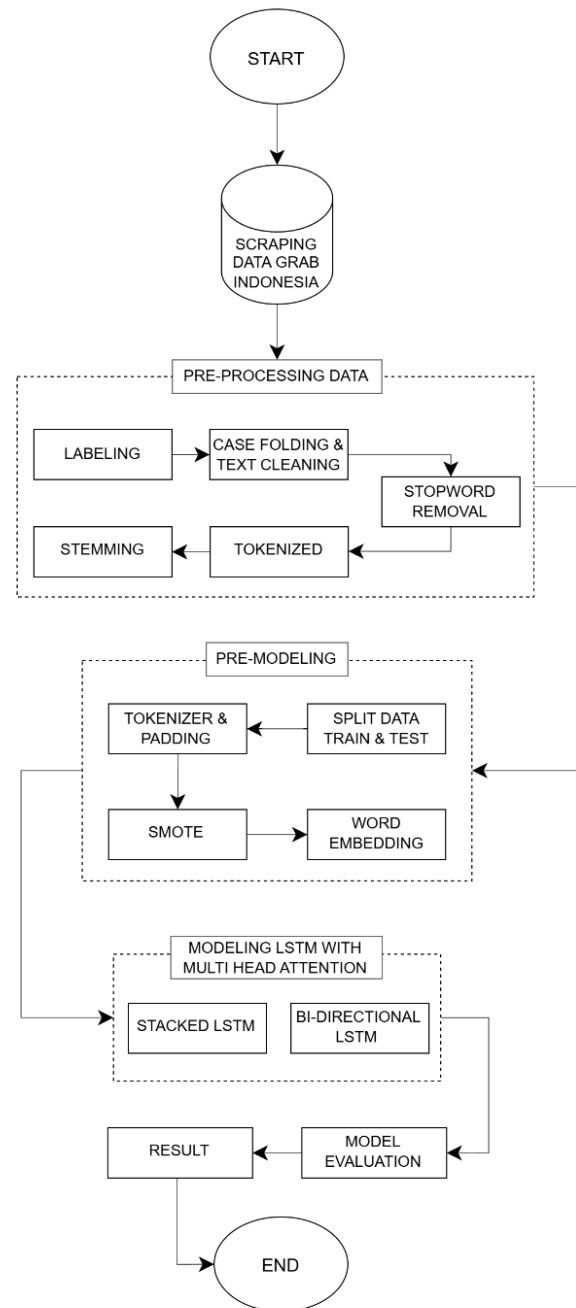
II. METODE PENELITIAN

Berikut adalah alur penelitian yang dilakukan pada penelitian ini, dapat dilihat pada Gambar 1.

A. Scraping Data

Data yang akan digunakan pada penelitian ini merupakan data ulasan pengguna aplikasi *Grab* Indonesia yang di ambil dengan cara melakukan scraping data di *google play store*

grab Indonesia pada tanggal 26 Juni 2024 diambil secara acak dan bertahap dengan total 2000 data [5].



Gambar 1. Alur Metode Penelitian

B. Pre-Processing

1) Labeling

Setiap ulasan diberikan label sentimen, yaitu positif (1) atau negatif (0), berdasarkan *rating* atau penilaian yang diberikan oleh pengguna. Ulasan dengan skor di bawah 3 dilabel sebagai negatif, sedangkan ulasan dengan skor lebih dari 3 dilabel sebagai positif. Label ini berfungsi sebagai target atau *ground truth* yang akan dipelajari oleh model

selama proses pelatihan, untuk mengenali pola sentimen dalam data teks [5].

2) Case Folding & Text Cleaning

Case Folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil untuk mengurangi keragaman kata yang disebabkan oleh perbedaan dalam penggunaan huruf kapital. *Text cleaning* melibatkan penghapusan karakter yang tidak diinginkan seperti tanda baca, angka, dan simbol yang tidak memiliki arti dalam analisis sentimen. Langkah ini bertujuan untuk membersihkan teks sehingga lebih mudah diproses oleh model [6].

3) Stopword Removal

Stopwords adalah kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan banyak informasi dalam analisis, seperti "dan", "atau", "adalah". Menghapus *stopwords* dapat membantu meningkatkan kinerja model dengan mengurangi jumlah kata yang akan diproses [7].

4) Tokenizing

Tokenizing adalah proses membagi teks menjadi unit-unit yang lebih kecil yang disebut token, seperti kata atau frasa. Langkah ini membuat data teks lebih mudah diproses oleh model, karena data akan diubah menjadi struktur yang lebih sederhana untuk dianalisis [8].

5) Stemming

Stemming merupakan proses mengubah kata yang memiliki imbuhan menjadi bentuk dasarnya. Sebagai contoh, kata "berlari" akan diubah menjadi "lari". Proses ini membantu mengurangi variasi kata yang muncul dan menjadikan analisis lebih konsisten, karena model akan menganggap kata dengan bentuk dasar yang sama sebagai satu entitas [7].

C. Pre-Modeling

1) Split Data Train & Test

Dataset dibagi menjadi dua bagian utama, yaitu data latih (train) dan data uji (test). Data latih digunakan untuk mengajarkan model mengenali pola-pola dalam data dan memahami hubungan antar fitur. Di sisi lain, data uji berfungsi untuk mengukur performa model dengan menggunakan data baru yang belum pernah dilihat sebelumnya [9].

2) Tokenizing & Padding

Tokenisasi dan *padding* adalah dua tahapan penting dalam proses persiapan data teks untuk analisis sentimen. Tokenisasi mengonversi teks mentah menjadi urutan angka dengan memberikan indeks pada setiap kata berdasarkan frekuensi kemunculannya, di mana kata-kata yang tidak ditemukan dalam kosakata akan digantikan dengan token khusus seperti $\langle OOV \rangle$. Sementara itu, *padding* bertujuan untuk menyamakan panjang setiap urutan token dengan menambahkan angka nol di awal atau akhir urutan, sehingga memastikan semua input memiliki ukuran yang konsisten dan memungkinkan pemrosesan *batch* yang efisien oleh model seperti LSTM [10].

3) Smote

Smote (Synthetic Minority Over-sampling Technique) merupakan teknik yang digunakan untuk mengatasi ketidakseimbangan kelas dalam dataset. Metode ini menghasilkan data sintetik baru untuk kelas minoritas, sehingga distribusi kelas menjadi lebih seimbang. Ketidakseimbangan data dapat menyebabkan model lebih memprioritaskan kelas mayoritas dan mengabaikan kelas minoritas, sehingga mengurangi akurasi prediksi. Dengan menggunakan *Smote*, distribusi kelas yang lebih seimbang dapat diperoleh, sehingga model dapat belajar secara optimal dari data yang tersedia [11].

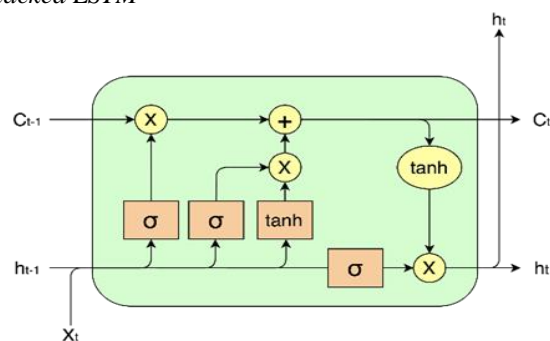
4) Word Embedding

Word Embedding merupakan proses untuk mengubah kata-kata dalam data teks menjadi representasi numerik dalam bentuk vektor. Teknik ini membantu model memahami hubungan semantik antar kata berdasarkan kedekatan vektor di ruang dimensi tertentu. *Word Embedding* dilakukan setelah proses tokenisasi dan padding untuk memastikan format data yang konsisten. Representasi numerik ini menjadi input untuk model, yang memungkinkan model memahami konteks serta hubungan antar kata secara lebih mendalam. Teknik ini penting untuk analisis sentimen karena dapat menangkap pola kompleks dalam data teks, sehingga meningkatkan akurasi prediksi [11].

D. Modeling Menggunakan LSTM

LSTM merupakan pengembangan dari recurrent neural network (*RNN*) yang hadir untuk mengatasi kelemahan yang dimiliki *RNN*. Salah satu masalah utama *RNN* adalah kesulitan dalam memprediksi data berdasarkan informasi yang telah disimpan dalam jangka waktu yang lama. *LSTM* dirancang dengan kemampuan utama untuk menyimpan informasi yang relevan dan menghapus informasi yang tidak relevan dari waktu ke waktu [12].

1) Stacked LSTM



Gambar 2. LSTM Architecture

Model LSTM yang akan digunakan adalah *Stacked LSTM* karena memanfaatkan beberapa layer untuk memodelkan data yang ada pada dataset ulasan pengguna *Grab* Indonesia, dikarenakan kemampuan model untuk mengatasi masalah *vanishing gradient* pada *RNN* dengan kemampuan untuk menyimpan informasi jangka panjang pada unit memori atau

sel status. *LSTM* dirancang menggunakan tiga gerbang utama untuk prosesnya yaitu, *Forget Gate*, *Input Gate* dan *Output Gate*. Mekanisme proses tersebut memungkinkan *LSTM* untuk menyimpan, memperbaharui atau membuang informasi dengan efisien, sehingga dapat menangkap depensi jangka panjang pada data sekuensial [13].

Forget gate merupakan gerbang utama yang berfungsi untuk menentukan informasi yang perlu dibuang atau dipertahankan dalam *cell state*. Proses ini akan mengambil fungsi *sigmoid*, dimana nilai 0 = membuang dan nilai 1 = menyimpan [14] [15].

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f)$$

f_t : Nilai dari *forget gate*

W_f : Bobot *forget gate*

h_{t-1} : *Output* dari langkah sebelumnya

X_t : *Input* saat ini

b_f : Bias

Input Gate merupakan gerbang kedua yang bertugas untuk menambahkan informasi baru kedalam *cell state*. Dalam proses ini terdapat 2 tahap yaitu, fungsi *sigmoid* untuk memilih informasi yang dimasukkan dan *tanh* untuk membuat skala dari informasi tersebut [14][15].

Nilai tersebut diperbarui dengan rumus berikut:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i)$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c)$$

Jika nilai telah selesai diperbarui selanjutnya nilai *cell state* akan diperbaharui dengan rumus:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t$$

i_t : Nilai dari *input gate*

C_t : Nilai kandidat *cell state*

$W_i W_c$: Bobot *input gate* dan kandidat

b_i, b_c : Bias

Output Gate, proses terakhir dimana nantinya akan dilewati oleh *cell state* setelah dilakukan pembaruan *cell* [14][15]. Fungsi *sigmoid* akan digunakan untuk memilih bagian informasi yang akan diambil sebagai *output* kemudian hasilnya akan dinormalisasi dengan fungsi *tanh*, dengan rumus berikut:

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

o_t : Nilai *Output Gate*

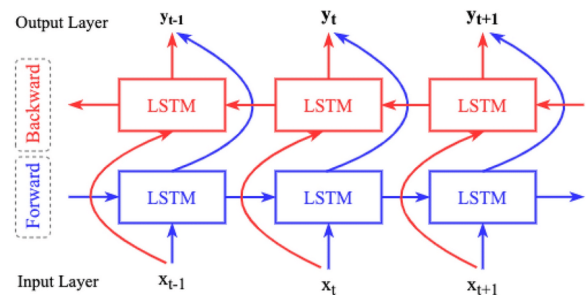
h_t : *Output* pada langkah waktu saat ini

W_o : Bobot *output gate*

b_o : Bias

2) *Bi-Directional LSTM*

BiLSTM merupakan pengembangan dari model *Lstm* biasa yang berfungsi untuk mengatasi kekurangan dari *Lstm* yang hanya memproses data secara satu arah. *BiLSTM* bekerja dengan menjalankan dua unit *LSTM* secara bersamaan, yaitu proses arah maju (*forward*) dan arah mundur (*backward*). Dengan proses tersebut *BiLSTM* dapat bekerja dengan baik untuk menangkap informasi dari kedua arah serta meningkatkan kemampuan model untuk memproses urutan data yang lebih kompleks [16][17].



Gambar 3. *Bi-Directional Lstm Architecture*

$$h_t \text{ bilstm} = h_t \text{ forward}, h_t \text{ backward}$$

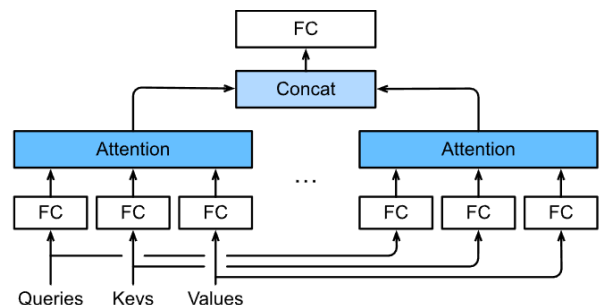
$h_t \text{ forward}$: Representasi *Lstm* saat membaca data dari awal ke akhir.

$h_t \text{ backward}$: Representasi *Lstm* saat membaca data dari akhir ke awal.

Kedua output tersebut digabungkan menjadi representasi akhir, yaitu $h_t \text{ bilstm}$. Dengan memanfaatkan proses data dua arah, *Bi-Directional LSTM* dapat memahami hubungan dengan lebih baik dibandingkan *LSTM* satu arah.

E. *Multi Head Attention*

Multi-Head Attention (MHAT) akan digunakan untuk meningkatkan kinerja dari dua model *LSTM* tersebut. *MHAT* dapat menangkap pola hubungan yang lebih kompleks dibandingkan dengan mekanisme *attention* tunggal. Dengan begitu, penerapan *MHAT* dalam kinerja model *LSTM* dan *BiLSTM* diharapkan dapat meningkatkan akurasi dan kemampuan model dalam memahami konteks data teks lebih baik lagi [18].



Gambar 4. *Multi Head Attention Architecture*

$$\text{MultiHead}(Q, K, V = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^0$$

Gambar 4 diatas model *Multi-Head Attention (MHAT)* memproses data Q (*Query*), K (*Key*), dan V (*Value*), data tersebut diolah dengan bobot W^Q, W^K, W^V untuk setiap "head". Setiap head menghitung seberapa relevan nilai antara Q dan K menggunakan perkalian *dot-product*, kemudian hasilnya diubah menjadi probabilitas dengan softmax. Nilai ini digunakan untuk memilih informasi penting dari V . Output dari setiap head digabungkan menjadi satu melalui operasi *concatenation*, lalu diolah lagi dengan matriks W^0 untuk menghasilkan representasi akhir yang lebih kaya. Mekanisme ini memungkinkan model fokus pada berbagai aspek data secara bersamaan, sehingga lebih efektif dalam memahami konteks untuk menghasilkan output perhatian.

F. Model Evaluation

Evaluasi model dilakukan untuk menilai kinerja dari *LSTM* dan *BiDirectional LSTM* yang sudah dioptimasi menggunakan *MHAT* dalam melakukan analisis sentimen pada data ulasan layanan aplikasi grab. Untuk menilai hasil dari kinerja model pada penilitan ini, digunakan *confusion matrix* seperti yang bisa dilihat pada TABEL I dibawah untuk melihat performa model klasifikasi dengan detail pada prediksi terkait label kelas positif dan negatif [19].

TABEL I
3 TERATAS ISI DATASET

Confusion Matrix	Prediction Positive	Prediction Negative
Actual Positive	TP	FP
Actual Negative	FN	TN

Selama proses ini, peneliti akan mengevaluasi kinerja dari empat elemen, yaitu nilai *True Positive (TP)*, nilai *True Negative (TN)*, nilai *False Positive (FP)* dan nilai *False Negative (FN)*, menggunakan rumus perhitungan berikut:

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

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

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

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

Untuk menentukan model mana yang memiliki kinerja lebih baik, hasil perbandingan model *Stacked LSTM* dan *Bi-Directional LSTM* akan dilakukan setelah hasil dari semua elemen telah diperoleh, mulai dari *accuracy*, *precision*, *recall* dan *f1-score* [20][21].

III. HASIL DAN PEMBAHASAN

Pada pembahasan hasil penelitian ini dimulai dengan pengambilan dataset seperti yang dapat dilihat pada TABEL II dan selanjutnya data yang sudah didapatkan akan diolah terlebih dahulu sebelum dilakukan proses modeling agar hasil yang didapatkan lebih optimal.

A. Scraping Data

TABEL II
3 TERATAS ISI DATASET

No	Score	Content
1	5	Guah pelanggan GRAB dari pertama ada GRAB,, dan Gua udh lama pakai apk ini, dari zaman AZINAMOTO menurut gua dari semua banyak aplikasi penjemputan manusia, yang paling bagus ya GRAB jawabannya,, yang bilang GRAB buruk,, ya ELU nya aja kudet,,
2	4	Dapat dibbilang saya sangat bergantung pada Grab ketika ingin pesan makanan, nitip beli obat, maupun bepergian. Oleh karena itu, fitur jumlah saldo OVO yg lgsg terlihat di Home sangat saya perlukan. Sayang sekali, fitur tersebut skrg dihilangkan dan membuat saya harus klik Akun >> Metode Pembayaran hanya utk melihat berapa sisa saldo aktif saya.
3	3	Titik suka tidak sesuai padahal sudah saya sesuaikan dengan titik biru, terkadang titik tujuan tidak sama dengan Google Map, map pada Grab tidak seakurat Google Map jadi suka salah penjemputanya/ tujuannya. Tolong untuk diperbaiki lagi🙏

Proses pengambilan dataset dilakukan dengan melakukan scraping data ulasan dari aplikasi *Grab* yang tersedia di *Google Play Store* secara acak dan bertahap pada tanggal 23 Juni 2024. Data diambil berdasarkan isi ulasan berbahasa Indonesia dan negara pengguna yaitu Indonesia, kemudian diambil berdasarkan kata kunci "MOST-RELEVANT" untuk ulasan yang relevan, panjang, informatif, serta mencerminkan pengalaman nyata pengguna mengenai pengalaman menggunakan aplikasi *Grab*, dengan total 2000 ulasan. Skor/rating ulasan difilter dengan nilai *None*, yang berarti mengambil seluruh isi dari kolom ulasan mulai dari skor bintang 1 hingga 5 diambil tanpa terkecuali.

B. Pre-Processing

1) Labeling

Proses pelabelan dilakukan untuk mengelompokkan sentimen ulasan pengguna aplikasi *Grab* Indonesia berdasarkan nilai skor pada kolom yang tersedia. Ulasan dengan skor di atas 3 akan diberi label 1 (positif), sementara ulasan dengan skor kurang atau sama dengan 3 akan diberi label 0 (negatif). Selain itu, dilakukan juga pengecekan manual untuk memastikan bahwa isi ulasan yang dipilih benar-benar mencerminkan sentimen yang sesuai, baik itu positif maupun negatif dengan hasil yang dapat dilihat pada Tabel III dibawah. Tahapan ini sangat penting untuk menjaga

kualitas data dan meningkatkan akurasi model dalam menganalisis sentiment.

TABEL III
LABELING

No	Score	Content	label
1	5	Guah pelanggan GRAB dari pertama ada GRAB,,, dan Gua udh lama pakai apk ini, dari zaman AZINAMOTO menurut gua dari semua banyak aplikasi penjemputan manusia, yang paling bagus ya GRAB jawabannya,,, yang bilang GRAB buruk,,, ya ELU nya aja kudet,,,	1
2	4	Dapat dibilang saya sangat bergantung pada Grab ketika ingin pesan makanan, nitip beli obat, maupun bepergian. Oleh karena itu, fitur jumlah saldo OVO yg lgsg terlihat di Home sangat saya perlukan. Sayang sekali, fitur tersebut skrg dihilangkan dan membuat saya harus klik Akun >> Metode Pembayaran hanya utk melihat berapa sisa saldo aktif saya.	1
3	4	Titik suka tidak sesuai padahal sudah saya sesuaikan dengan titik biru, terkadang titik tujuan tidak sama dengan Google Map, map pada Grab tidak seakurat Google Map jadi suka salah penjemputanya/ tujuannya. Tolong untuk diperbaiki lagi🙏	0

Setelah dilakukan labeling didapatkan data sentimen positif sebesar 820 ulasan dan sentimen negatif sebesar 1180 ulasan.

2) *Case Folding & Text Cleaning*

Case Folding dilakukan untuk mengubah semua teks menjadi huruf kecil untuk menghilangkan perbedaan yang disebabkan oleh penggunaan huruf kapital, kemudian *text cleaning* diterapkan untuk menghapus karakter-karakter yang tidak relevan seperti tanda baca, angka, dan simbol, sehingga teks menjadi lebih bersih dan mudah diproses, Hasilnya dapat dilihat pada Tabel V dibawah ini.

TABEL IV
SEBELUM DITERAPKAN CASEFOLDING & TEXT CLEANING

Content
Aplikasinya bagus dan tampilan antarmukanya juga bagus. Tidak ada kendala sama sekali dalam menggunakan aplikasi ini. Saya sangat menyukainya pada saat memesan makanan, karena ongkirnya murah dan terdapat berbagai pilihan pengantaran dengan biaya yang berbeda dan waktu pengiriman yang berbeda juga. Drivernya juga ramah.

TABEL V
SESUDAH DITERAPKAN CASEFOLDING & TEXT CLEANING

Content
aplikasinya bagus dan tampilan antarmukanya juga bagus tidak ada kendala sama sekali dalam menggunakan aplikasi ini saya

sangat menyukainya pada saat memesan makanan karena ongkirnya murah dan terdapat berbagai pilihan pengantaran dengan biaya yang berbeda dan waktu pengiriman yang berbeda juga drivernya juga ramah

3) *Stopwords Removal*

Langkah selanjutnya adalah menghapus *stopwords*, yaitu kata-kata yang sering muncul dalam teks namun tidak memberikan informasi penting, dengan contoh kata seperti "dan", "yang", "dengan" dan "adalah". Penghilangan *stopwords* ini penting untuk meningkatkan efisiensi pemrosesan dan kualitas analisis, karena dengan menghapus kata-kata yang tidak relevan, model dapat lebih fokus pada kata-kata yang memiliki makna substansial, contohnya dapat dilihat pada Tabel VI. Hal ini membantu dalam mengurangi dimensi data yang harus dianalisis, mempercepat proses komputasi, dan meningkatkan akurasi model dalam mengekstrak informasi yang signifikan untuk analisis sentimen atau aplikasi pengolahan bahasa alami lainnya.

TABEL VI
STOPWORDS REMOVAL

Content
['aplikasinya', 'bagus', 'tampilan', 'antarmukanya', 'juga', 'bagus', 'tidak', 'ada', 'kendala', 'sama', 'sekali', 'dalam', 'menggunakan', 'aplikasi', 'ini', 'saya', 'sangat', 'menyukainya', 'saat', 'memesan', 'makanan', 'karena', 'ongkos', 'murah', 'terdapat', 'banyak', 'pilihan', 'pengantaran', 'biaya', 'berbeda', 'waktu', 'pengiriman', 'berbeda', 'juga', 'driver', 'juga', 'ramah']

4) *Tokenized*

TABEL VII
TOKENIZED

Content
['aplikasinya', 'bagus', 'dan', 'tampilan', 'antarmukanya', 'juga', 'bagus', 'tidak', 'ada', 'kendala', 'sama', 'sekali', 'dalam', 'menggunakan', 'aplikasi', 'ini', 'saya', 'sangat', 'menyukainya', 'pada', 'saat', 'memesan', 'makanan', 'karena', 'ongkos', 'murah', 'dan', 'terdapat', 'banyak', 'pilihan', 'pengantaran', 'dengan', 'biaya', 'yang', 'berbeda', 'dan', 'waktu', 'pengiriman', 'yang', 'berbeda', 'juga', 'driver', 'juga', 'ramah']

Setelah teks selesai dibersihkan, langkah berikutnya adalah melakukan tokenisasi (*tokenizing*), yaitu proses memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau frasa, yang disebut sebagai token, seperti yang ada di Tabel VII diatas. Tokenisasi membantu dalam menyusun teks menjadi struktur yang lebih terorganisir, sehingga memudahkan analisis lebih lanjut. Proses ini juga mempersiapkan teks untuk tahap representasi numerik, seperti word embedding, yang diperlukan agar model dapat memahami data dengan lebih baik dan menghasilkan prediksi yang lebih akurat.

5) *Stemming*

Setelah menghapus *stopwords*, tahap selanjutnya adalah melakukan *stemming*, yang berfungsi untuk mengubah kata-kata dengan imbuhan menjadi bentuk dasar. Contohnya, kata "menilai" akan disederhanakan menjadi "nilai". *Stemming*

sangat berguna untuk mengurangi variasi kata yang tidak perlu, sehingga analisis menjadi lebih terfokus dan konsisten. Hasil stemming dapat dilihat pada Tabel VIII. Selain itu, *stemming* membantu mengurangi kompleksitas data teks, yang memungkinkan model *machine learning* dalam analisis sentimen bekerja lebih efisien. Dengan kata dasar, jumlah fitur yang perlu dianalisis menjadi lebih sedikit, sehingga mempercepat pemrosesan data dan meningkatkan kemampuan model dalam memahami isi dari ulasan teks.

TABEL VIII
STEMMING

Content
['aplikasi', 'bagus', 'tampil', 'antarmuka', 'juga', 'bagus', 'tidak', 'ada', 'kendala', 'sama', 'sekali', 'dalam', 'guna', 'aplikasi', 'ini', 'saya', 'sangat', 'suka', 'saat', 'mes', 'makan', 'karena', 'ongkos', 'murah', 'dapat', 'banyak', 'pilih', 'antar', 'biaya', 'beda', 'waktu', 'kirim', 'beda', 'juga', 'driver', 'juga', 'ramah']

C. Pre-Modeling

1) Split Data Train & Test



Gambar 3. Split Data

Setelah data berhasil melalui tahap pembersihan sehingga menjadi dataset yang bersih dan siap digunakan, langkah selanjutnya adalah memasuki tahap *pre-modeling*. Pada tahap ini, dataset dibagi menjadi dua bagian, yaitu data latih (*train*) dan data uji (*test*), dengan proporsi 80:20 dari total 2000 data. Pembagian ini menghasilkan 1600 data untuk melatih model dan 400 data untuk menguji kinerjanya. Data latih digunakan untuk melatih model mengenali pola-pola dalam data, sementara data uji digunakan untuk mengevaluasi seberapa baik model bekerja pada data baru. Dengan pembagian ini, proses pelatihan dapat berjalan lebih optimal sehingga hasil dari evaluasi model yang akan dilakukan memiliki kemampuan generalisasi yang baik.

2) Tokenizer & Padding

Setelah data sudah di bagi menjadi data latih (*train*) dan data uji (*test*), selanjutnya di lakukan tokenisasi yang

bertujuan untuk mengubah teks menjadi format numerik, di mana setiap kata dalam teks diwakili oleh indeks unik. Hal ini memungkinkan teks dipahami oleh model berbasis numerik. Setelah tokenisasi, dilakukan padding untuk memastikan bahwa setiap urutan token memiliki panjang yang sama dengan menambahkan nilai nol (0) di akhir urutan yang lebih pendek. Padding ini penting agar model dapat memproses data secara keseluruhan tanpa masalah perbedaan panjang urutan, ketiga proses tersebut dapat dilihat pada Tabel IX sampai Tabel XI berikut.

TABEL IX
TEXT

Content
['aplikasi', 'bagus', 'tampil', 'antarmuka', 'juga', 'bagus', 'tidak', 'ada', 'kendala', 'sama', 'sekali', 'dalam', 'guna', 'aplikasi', 'ini', 'saya', 'sangat', 'suka', 'saat', 'mes', 'makan', 'karena', 'ongkos', 'murah', 'dapat', 'banyak', 'pilih', 'antar', 'biaya', 'beda', 'waktu', 'kirim', 'beda', 'juga', 'driver', 'juga', 'ramah']

TABEL X
TOKENIZING

Content
[4, 8, 113, 114, 14, 8, 5, 15, 115, 37, 34, 39, 40, 4, 17, 2, 10, 12, 116, 117, 28, 30, 118, 119, 27, 18, 41, 120, 121, 42, 122, 123, 42, 14, 9, 14, 124]

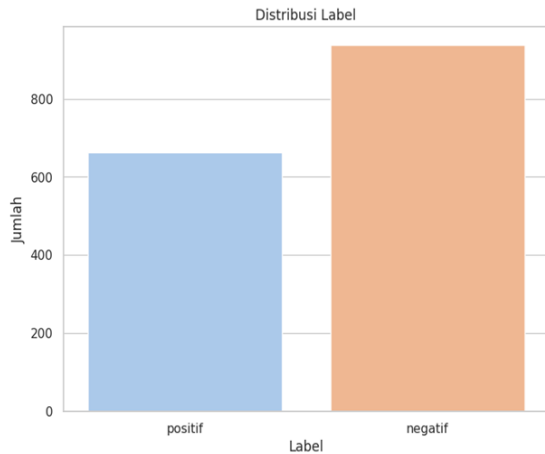
TABEL XI
PADDING

Content
[4 8 113 114 14 8 5 15 115 37 34 39 40 4 17 2 10 12 116 117 28 30 118 119 27 18 41 120 121 42 122 123 42 14 9 14 124 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 00]

Dilakukan padding dengan menambahkan nilai nol (0) di akhir urutan token hingga mencapai panjang maksimum 100. Jika panjang urutan melebihi batas maksimum, bagian akhir dari urutan akan dipotong sesuai parameter *truncating='post'*. Pada hasil padding, urutan token asli tetap berada di awal, sedangkan nilai nol ditambahkan di bagian akhir untuk melengkapi panjangnya hingga 100.

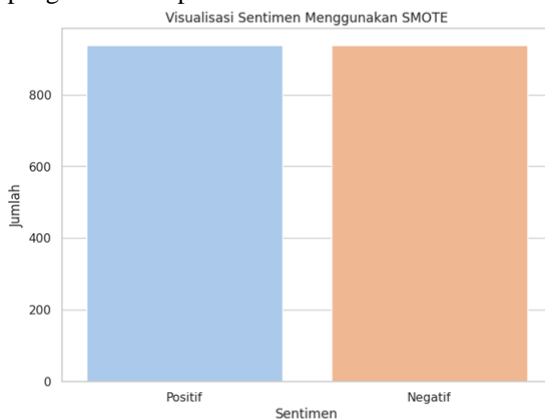
3) Smote Data

Setelah melewati proses tokenisasi dan *padding*, selanjutnya adalah menerapkan teknik *SMOTE* (*Synthetic Minority Over-sampling Technique*) untuk menyeimbangkan jumlah label positif dan negatif pada data yang sudah ditokenisasi dan padding sebelumnya sebelum masuk kedalam proses modeling.



Gambar 4. Data Sebelum Dilakukan Smote

Sebelum diterapkan teknik SMOTE, data pelatihan mendapatkan hasil 1600 data dengan jumlah label positif sebanyak 663 dan label negatif sebanyak 937. Teknik SMOTE digunakan untuk mengatasi ketidakseimbangan kelas dalam dataset dengan meningkatkan jumlah data pada kelas minoritas. Ketika kelas tidak seimbang, model cenderung memprediksi kelas mayoritas secara berlebihan, yang dapat mempengaruhi hasil prediksi.



Gambar 5. Data Sesudah Dilakukan Smote

Setelah menerapkan teknik SMOTE, jumlah label sentimen positif dan negatif menjadi seimbang, yaitu dengan hasil 937 data untuk kedua label. Proses SMOTE berhasil dilakukan dengan menambah jumlah data pada kelas minoritas (label positif) sehingga distribusi antara kelas positif dan negatif kini seimbang. Dengan distribusi yang seimbang, model dapat dilatih dengan lebih baik tanpa adanya kecenderungan bias data pada kelas mayoritas.

4) Word Embedding

Setelah data latih (*train*) telah berhasil di seimbangkan, kemudian akan dilanjutkan kedalam proses layer embedding, dimana data yang sudah di tokenisasi dan padding sebelumnya akan diubah lebih kecil lagi menjadi *vector* numerik. Dimensi embedding yang di gunakan = 100, dimana setiap kata dalam data akan diubah menjadi vektor numerik berdimensi 100, kemudian digunakan $max_word = 10.000$

yang akan memprioritaskan kata-kata yang paling sering muncul sebanyak 10.000 kata dan yang terakhir menggunakan panjang *sequence* (kalimat) dengan maksimal 100 token, jika sebuah kalimat lebih pendek dari 100 token maka *padding* akan menambahkan nilai nol (0) hingga sisa panjang tokennya terpenuhi, hal tersebut dilakukan agar panjang kalimat tetap seimbang sehingga proses modeling berjalan dengan baik

D. Modeling LSTM

1) Stacked LSTM

Proses dimulai dengan *layer embedding*, yang mengubah kata-kata menjadi *vector* berdimensi 100. Selanjutnya, informasi sekuensial diproses melalui dua lapisan LSTM. Lapisan pertama memiliki 128 *unit*, diikuti oleh lapisan kedua dengan 128 *unit* juga. Untuk mencegah *overfitting*, *Dropout layer* diterapkan setelah setiap lapisan LSTM.

Setelah itu, model memanfaatkan *Multi-Head Attention* dengan 4 "head" untuk menangkap konteks yang lebih kompleks dalam data sekuensial. *Output* dari *Attention* ini digabungkan dengan *output LSTM* kedua menggunakan *Add layer* dan dinormalisasi melalui *LayerNormalization*.

Selanjutnya, *Global Average Pooling* digunakan untuk mereduksi dimensi *output* menjadi satu nilai per fitur. *Dropout* kembali diterapkan untuk meningkatkan generalisasi model sebelum akhirnya melewati *Dense layer* yang menghasilkan *output biner* (positif/negatif).

TABEL XII
TRAINING DATA

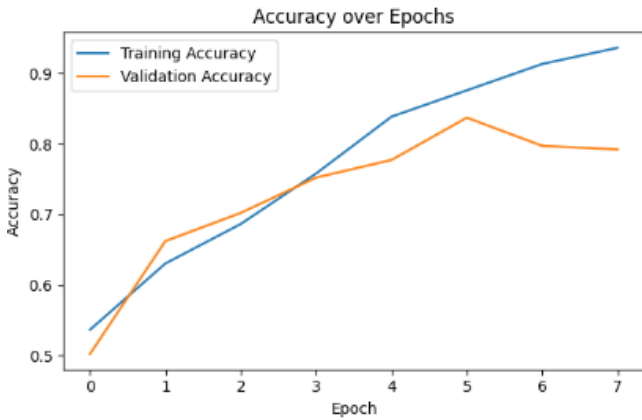
Epoch	Akurasi Training	Akurasi Validasi	Loss Training	Loss Validasi
1	51,51%	50,25%	0,8927	0,7621
2	62,96%	66,25%	0,6591	0,6132
3	68,78%	70,25%	0,6079	0,5560
4	74,41%	75,25%	0,5224	0,4913
5	84,05%	77,75%	0,3838	0,5295
6	88,17%	83,75%	0,3023	0,4727
7	91,83%	79,75%	0,2326	0,5468
8	94,55%	79,25%	0,1865	0,6880

Perhitungan *epoch* dilakukan sampai 8 percobaan dengan hasil akurasi terbaik tercatat pada *epoch* ke-6 dengan Akurasi Validasi 83,75% dan *Loss* Validasi 0,4727, seperti yang bisa dilihat pada TABEL XII. Pada *epoch* ke-6, model mencapai akurasi validasi tertinggi, yang menyebabkan *EarlyStopping* menghentikan pelatihan dengan *limit* batasan 2 percobaan dan mengembalikan bobot model kepada hasil terbaik.

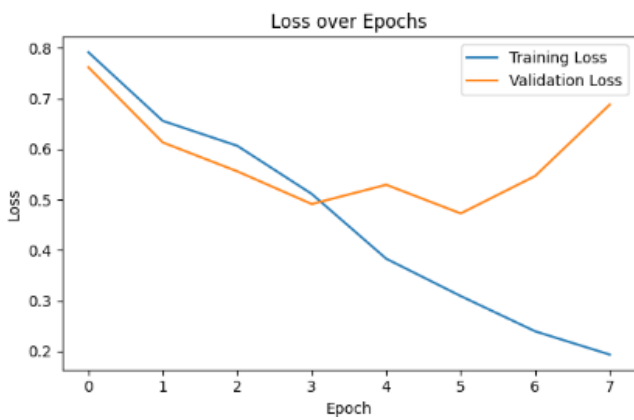
Akurasi Terakhir (Setelah *EarlyStopping*):

Akurasi Validasi: 83,75%

Loss Validasi: 0,4727



Gambar 6. Tingkat Akurasi dari Epoch

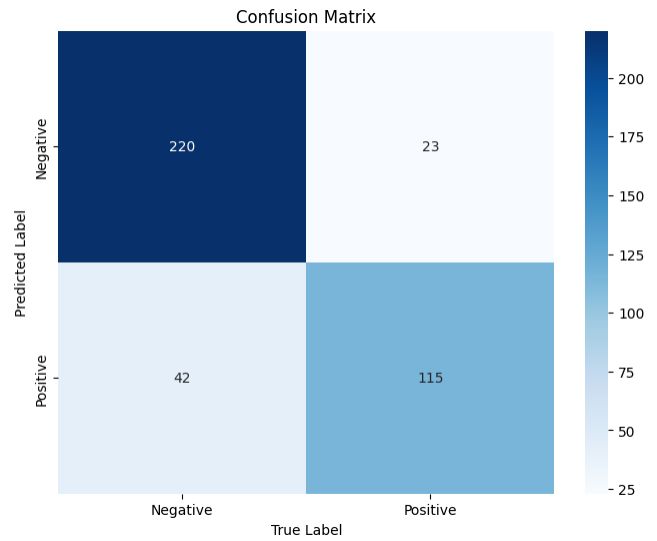


Gambar 7. Tingkat Loss dari Epoch

Dari hasil yang didapatkan pada Gambar 4.6 dan 4.7, model terindikasi mengalami *overfitting* dikarenakan akurasi data pelatihan terus meningkat sementara data validasi mengalami penurunan setelah epoch ke-6. Untuk mengatasi masalah ini, "*EarlyStopping*" diterapkan guna menghentikan pelatihan dan mengembalikan model ke epoch ke-6, di mana akurasi validasi masih optimal. Ini memastikan generalisasi yang lebih baik pada data percobaan dan mencegah model terlalu menyesuaikan diri dengan data pelatihan.

TABEL XIII
TRAINING DATA

Label	Precision	Recall	F1-Score	Support
Negative	0,84	0,91	0,87	243
Positive	0,83	0,73	0,78	157
Accuracy			0,84	400
Macro avg	0,84	0,82	0,83	400
Weighted avg	0,84	0,84	0,84	400



Gambar 8. Confusion Matrix Stacked LSTM

Dari hasil evaluasi model yang dapat dilihat pada Tabel XIII dan Gambar 8 diatas, model *STACKED LSTM* berhasil memprediksi 220 kasus negatif dengan benar (*True Negative*) dan 115 kasus positif dengan benar (*True Positive*). Namun, terdapat 23 kasus negatif yang salah diprediksi sebagai positif (*False Positive*) dan 42 kasus positif yang salah diprediksi sebagai negatif (*False Negative*). Akurasi keseluruhan model mencapai 84%, dengan *precision* untuk kelas negatif sebesar 0,84 dan *recall* sebesar 0,91, sedangkan untuk kelas positif *precision* adalah 0,83 dan *recall* 0,73. *F1-score* untuk kelas negatif adalah 0,87 dan untuk kelas positif adalah 0,78. Ini menunjukkan keseimbangan yang cukup baik antara *precision* dan *recall* pada kedua kelas.

2) Bi-Directional LSTM

Proses dimulai dengan *Embedding Layer*, yang mengubah data input berupa indeks *integer* dengan panjang maksimum 100 menjadi representasi *vector* berdimensi 100. Representasi ini kemudian diproses oleh *Bidirectional LSTM*, yang menangkap konteks dari kedua arah (maju dan mundur) dengan *output* berdimensi 256, sehingga memberikan pemahaman yang lebih kaya terhadap urutan data.

Selanjutnya, model memanfaatkan *Multi-Head Attention* dengan 4 "*head*" untuk memahami hubungan kompleks antar kata dalam data sekuensial. Hasil dari mekanisme ini kemudian distabilkan melalui *Layer Normalization*, yang memastikan pelatihan berjalan dengan stabil. *Global Average Pooling* digunakan untuk mereduksi dimensi *output* menjadi rata-rata per fitur, yang kemudian dilanjutkan dengan *Dropout* untuk mencegah *overfitting* dan meningkatkan kemampuan generalisasi model.

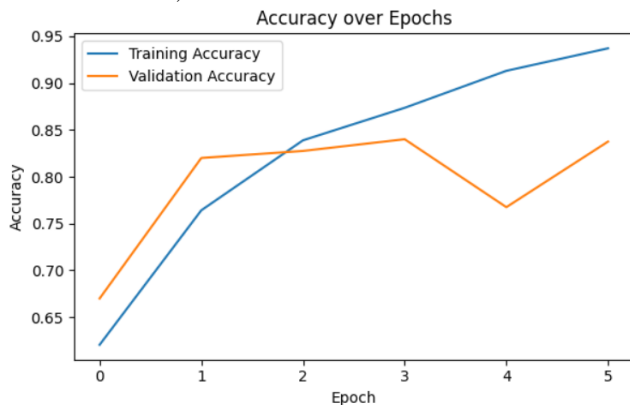
Pada tahap akhir, hasilnya diteruskan ke *Dense Layer* dengan fungsi aktivasi *sigmoid*, yang mengubahnya menjadi probabilitas untuk klasifikasi *biner* (positif/negatif).

TABEL XIV
TRAINING DATA

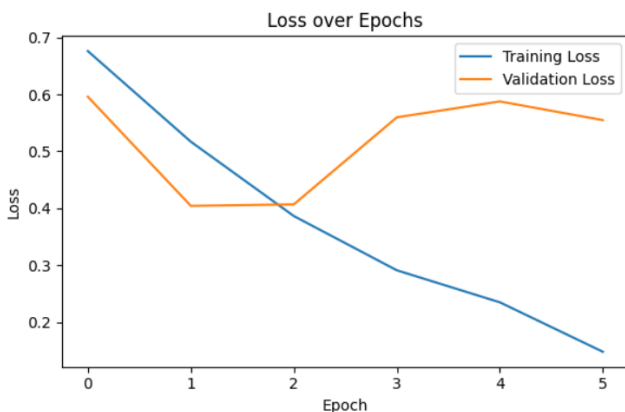
Epoch	Akurasi Training	Akurasi Validasi	Loss Training	Loss Validasi
1	52,77%	55,25%	0,8265	0,6734
2	70,78%	79,25%	0,5565	0,4691
3	82,73%	87,00%	0,3980	0,4062
4	84,74%	86,50%	0,3865	0,4472
5	87,47%	78,25%	0,3465	0,4938

Perhitungan epoch dilakukan sampai 5 percobaan dengan hasil akurasi terbaik tercatat pada epoch ke-3 dengan Akurasi Validasi 87.00% dan Loss Validasi 0. 4062 seperti yang bisa dilihat pada TABEL XIV. Pada epoch ke-3, model mencapai akurasi validasi tertinggi, yang menyebabkan *EarlyStopping* menghentikan pelatihan dengan *limit* batasan 2 percobaan dan mengembalikan bobot model kepada hasil terbaik.

Akurasi Terakhir (Setelah *EarlyStopping*):
 Akurasi Validasi: 87,0%
 Loss Validasi: 0,4062



Gambar 9. Tingkat Akurasi dari Epoch



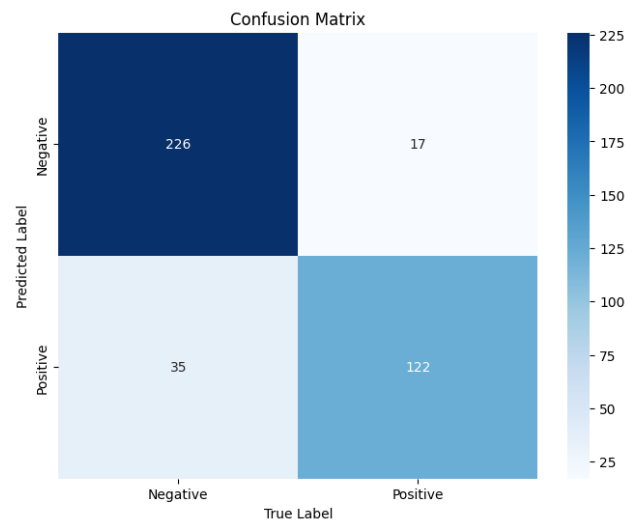
Gambar 10. Tingkat Loss dari Epoch

Dari hasil yang didapatkan pada Gambar 4.9 dan 4.10, model terindikasi mengalami *overfitting* dikarenakan akurasi data pelatihan terus meningkat sementara data validasi mulai stagnan dan menurun setelah epoch ke-3. Untuk mengatasi masalah ini, "*EarlyStopping*" diterapkan guna menghentikan pelatihan dan mengembalikan model ke

epoch ke-3, di mana akurasi validasi masih optimal. Ini memastikan generalisasi yang lebih baik pada data test dan mencegah model terlalu menyesuaikan diri dengan data pelatihan.

TABEL XV
TRAINING DATA

Label	Precision	Recall	F1-Score	Support
Negative	0,87	0,93	0,90	243
Positive	0,88	0,78	0,82	157
Accuracy			0,87	400
Macro avg	0,87	0,85	0,86	400
Weighted avg	0,87	0,87	0,87	400



Gambar 11. Confusion Matrix Bi-Directional LSTM

Hasil evaluasi model *Bi-Directional LSTM* yang dapat dilihat pada TABEL XV dan Gambar 11 diatas, menunjukkan peningkatan performa dibandingkan dengan hasil evaluasi *Stacked LSTM*. Model berhasil memprediksi 226 kasus negatif dengan benar (*True Negative*) dan 122 kasus positif dengan benar (*True Positive*). Jumlah kesalahan prediksi berkurang, dengan hanya 17 kasus negatif yang salah diprediksi sebagai positif (*False Positive*) dan 35 kasus positif yang salah diprediksi sebagai negatif (*False Negative*). Akurasi keseluruhan meningkat menjadi 87%, dibandingkan dengan akurasi sebelumnya sebesar 84%. Untuk kelas negatif, *precision* meningkat menjadi 0,87, dan *recall* meningkat menjadi 0,93, menunjukkan kemampuan model yang lebih baik dalam mengidentifikasi kasus negatif. Pada kelas positif, nilai *precision* naik menjadi 0,88, sementara *recall* juga mengalami peningkatan menjadi 0,78. *F1-score* untuk kelas negatif juga meningkat dari 0,87 menjadi 0,90 sedangkan untuk kelas positif meningkat dari 0,78 menjadi 0,82. Secara keseluruhan, model menunjukkan peningkatan keseimbangan antara *precision* dan *recall*, dengan performa yang lebih baik dalam menghindari kesalahan prediksi pada kedua kelas. Performa ini menunjukkan bahwa model memiliki generalisasi yang lebih baik dibandingkan sebelumnya, dengan keunggulan signifikan pada kelas negatif.

IV. KESIMPULAN

Berdasarkan hasil penelitian mengenai analisis sentimen pada dataset ulasan pengguna aplikasi *Grab* Indonesia, dapat disimpulkan bahwa model *Bi-Directional LSTM* dengan penambahan mekanisme *Multi-Head Attention* menunjukkan performa yang lebih baik dibandingkan dengan model *Stacked LSTM*. *Bi-Directional LSTM*, yang memproses data sekuensial dengan mempertimbangkan konteks dari kedua arah, berhasil meningkatkan akurasi validasi dan mengurangi kesalahan prediksi, terutama pada kelas negatif. Model ini memiliki *precision* dan *recall* yang lebih tinggi untuk kelas negatif, yang menunjukkan kemampuannya yang lebih baik dalam mengidentifikasi sentimen negatif pengguna. Selain itu, penerapan teknik *SMOTE* untuk menangani tidak seimbang kelas juga berkontribusi pada hasil yang lebih seimbang antara kedua kelas.

Pada model *Bi-Directional LSTM*, akurasi validasi tertinggi tercatat sebesar 87%, dengan *F1-score* untuk kelas negatif mencapai 0.90 dan untuk kelas positif 0.82, menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall* pada kedua kelas. Sementara itu, model *Stacked LSTM* hanya mencapai akurasi 84%, dengan *F1-score* untuk kelas negatif 0.87 dan untuk kelas positif 0.78. Hasil ini menegaskan bahwa model *Bi-Directional LSTM* memiliki kemampuan yang lebih baik dalam memahami hubungan kontekstual dalam data ulasan yang bersifat kompleks dan sekuensial.

Secara keseluruhan, penelitian ini menunjukkan bahwa penambahan teknik *Multi-Head Attention* pada model *LSTM* khususnya *Bi-Directional LSTM* dapat menghasilkan model yang lebih efektif dalam menganalisis sentimen pengguna aplikasi *Grab*, dengan akurasi yang lebih tinggi dan kinerja yang lebih baik dalam mengklasifikasikan sentimen positif dan negatif dibandingkan dengan *Stacked LSTM*.

DAFTAR PUSTAKA

- [1] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 3, p. 1018, Jul. 2021, doi: 10.30865/mib.v5i3.3111.
- [2] D. Gibson Nababan *et al.*, "Merancang Analisis Sentimen Berdasarkan Pendapat Pengguna Aplikasi Grab Dengan Menggunakan Bahasa Pemrograman Python," 2024.
- [3] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, and R. Ocanitra, "Gojek and Grab User Sentiment Analysis on Google Play Using Naive Bayes Algorithm and Support Vector Machine Based Smote Technique," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012102.
- [4] D. R. Alghifari, M. Edi, and L. Firmansyah, "Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia," *J. Manaj. Inform.*, vol. 12, no. 2, pp. 89–99, Sep. 2022, doi: 10.34010/jamika.v12i2.7764.
- [5] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC," *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 42–48, 2023, doi: 10.54914/jtt.v9i1.609.
- [6] G. K. Locarso, "Analisis Sentimen Review Aplikasi Pedulilindungi Pada Google Play Store Menggunakan Nbc," *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 6, no. 2, pp. 353–361, 2022, doi: 10.59697/jtik.v6i2.207.
- [7] J. Nurvania, Jondri, and K. Muslim Lhaksamana, "Analisis Sentimen Pada Ulasan di TripAdvisor Menggunakan Metode Long Short-Term Memory (LSTM)," *e-Proceeding Eng.*, vol. 8, no. 4, pp. 4124–4135, 2021.
- [8] D. Sintia Amelia, N. Cahyana Aminuallah, and S. Informasi, "Teks Dan Analisis Sentimen Pada Chat Grup Whatsapp Menggunakan Long Short Term Memory (Lstm)," *J. Teknol. Terkini*, vol. 3, no. 2, p. 1, 2023, [Online]. Available: <http://teknologiterkini.org/index.php/terkini/article/view/354>
- [9] K. S. Witanto, N. A. Sanjaya ER, A. E. Karyawati, I. G. A. G. A. Kadyanan, I. K. G. Suhartana, and L. G. Astuti, "Implementasi LSTM Pada Analisis Sentimen Review Film Menggunakan Adam Dan RMSprop Optimizer," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 10, no. 4, p. 351, 2022, doi: 10.24843/jlk.2022.v10i04.p05.
- [10] E. I. Setiawan and I. Lestari, "Stance Classification Pada Berita Berbahasa Indonesia Berbasis Bidirectional LSTM," *J. Intell. Syst. Comput.*, vol. 3, no. 1, pp. 41–48, 2021, doi: 10.52985/insyst.v3i1.148.
- [11] M. H. Al-Areef and K. Saputra S, "Analisis Sentimen Pengguna Twitter Mengenai Calon Presiden Indonesia Tahun 2024 Menggunakan Algoritma LSTM," *J. SAINTIKOM (Jurnal Sains Manaj. Inform. dan Komputer)*, vol. 22, no. 2, p. 270, 2023, doi: 10.53513/jis.v22i2.8680.
- [12] A. D. Milniadi and N. O. Adiwijaya, "Analisis Perbandingan Model Arima Dan Lstm Dalam Peramalan Harga Penutupan Saham (Studi Kasus: 6 Kriteria Kategori Saham Menurut Peter Lynch)," *SIBATIK J. J. Ilm. Bid. Sos. Ekon. Budaya, Teknol. dan Pendidik.*, vol. 2, no. 6, pp. 1683–1692, 2023, doi: 10.54443/sibatik.v2i6.798.
- [13] M. Sharaf, E. E. D. Hemdan, A. El-Sayed, and N. A. El-Bahnasawy, "An efficient hybrid stock trend prediction system during COVID-19 pandemic based on stacked-LSTM and news sentiment analysis," vol. 82, no. 16. *Multimedia Tools and Applications*, 2023. doi: 10.1007/s11042-022-14216-w.
- [14] M. Z. Rahman, Y. A. Sari, and N. Yudistira, "Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 11, pp. 5120–5127, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [15] M. R. F. Kamarula and N. Rochmawati, "Perbandingan CNN dan Bi-LSTM pada Analisis Sentimen dan Emosi Masyarakat Indonesia Di Media Sosial Twitter Selama Pandemi Covid-19 yang Menggunakan Metode Word2vec," *J. Informatics Comput. Sci.*, vol. 04, pp. 219–228, 2022, doi: 10.26740/jinacs.v4n02.p219-228.
- [16] Ari Muzakir and Uci Suriani, "Model Deteksi Berita Palsu Menggunakan Pendekatan Bidirectional Long Short-Term Memory (BiLSTM)," *J. Comput. Inf. Syst. Ampera*, vol. 4, no. 2, pp. 93–105, 2023, [Online]. Available: <https://journal-computing.org/index.php/journal-cisa/article/view/397/185>
- [17] G. A. M. Kukuh Jaluwana, Gusti Made Arya Sasmita, and I Made Agus Dwi Suarjaya, "Analysis of Public Sentiment Towards Government Efforts to Break the Chain of Covid-19 Transmission in Indonesia Using CNN and Bidirectional LSTM," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 511–520, 2022, doi: 10.29207/resti.v6i4.4055.
- [18] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
- [19] R. Merdiansah and A. Ali Ridha, "Sentiment Analysis of Indonesian X Users Regarding Electric Vehicles Using IndoBERT," *J. Ilmu Komput. dan Sist. Inf. (JIKOMSI)*, vol. 7, no. 1, pp. 221–228, 2024.
- [20] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Inf. Process. Manag.*, vol. 58, no. 1, 2021, doi: 10.1016/j.ipm.2020.102435.
- [21] T. L. Nikmah, M. Z. Ammar, and Y. R. Allatif, "Comparison of LSTM, SVM, and naive bayes for classifying sexual harassment tweets," *J. Soft Comput. Explor.*, vol. 3, no. 2, pp. 131–137, 2022, doi: 10.52465/josce.v3i2.85.