

Metode *Long Short-Term Memory* dan *Lexicon Based* Untuk Analisis Sentimen Ulasan Aplikasi TikTok

Diny Wahyuni, Naufal Fadhillah dan Winda Widya Ariestya

Universitas Gunadarma

JL. Margonda Raya Depok Jawa Barat

E-mail : dwahyuni@staff.gunadarma.ac.id, naufalfadhillah2@gmail.com, winda_widya@staff.gunadarma.ac.id

Abstrak

Aplikasi TikTok menjadi salah satu aplikasi yang paling banyak di unduh yaitu lebih dari 1 miliar unduhan pada Google Playstore. Sebuah analisis sentimen diperlukan untuk mengetahui opini pengguna mengenai kepuasan pengguna dalam menggunakan aplikasi TikTok. Tahapan proses analisis sentimen dimulai dengan pengambilan data (crawling), text pre-processing, klasifikasi sentimen serta penyusunan hasil analisis sentimen. Hasil dari tahap text pre-processing yang diperoleh, dilakukan penentuan sentimen awal dengan menggunakan metode Lexicon Based dengan perhitungan otomatis. Selanjutnya dilakukan pembagian data menjadi data training dan data testing untuk mendapatkan model yang optimal dan memprediksi sentimen dengan model Long Short-Term Memory (LSTM). Pada penelitian ini ulasan dari sistem analisis sentimen dengan metode LSTM akurasi yang didapatkan sebesar 90,05%, precision 92,14%, recall 97,35%, dan F-1 Score 98,66%, ulasan positif 30,0%, ulasan negatif 59,5%, dan ulasan netral 10,5%. Hasil analisis sentimen pada penelitian ini menunjukkan bahwa aplikasi TikTok memiliki kecenderungan sentimen negatif pada saat data ulasan diambil.

Kata kunci : Analisis Sentimen, Long Short-Term Memory, Lexicon Based, Google Playstore, TikTok.

Pendahuluan

Pada umumnya analisis sentimen merupakan proses klasifikasi teks yang berfokus pada ulasan yang mengungkapkan apakah ulasan tersebut negatif atau positif dan analisis sentimen ini sangat berperan penting dalam memahami minat pengguna untuk membuat keputusan secepat mungkin [1]. Tujuan dari analisis sentimen adalah untuk mendapatkan informasi sentimen terhadap pengguna platform media sosial [2]. Berkaitan dengan media sosial, salah satu aplikasi yang saat ini sedang banyak mendapatkan perhatian adalah aplikasi TikTok. Aplikasi TikTok ini menjadi salah satu aplikasi yang paling banyak di unduh yaitu lebih dari 1 miliar unduhan pada Google Playstore, pengguna tersebut mayoritas dari anak usia sekolah, anak millennial atau generasi Z [3]. Penelitian ini dilakukan pada aplikasi Tiktok karena Tiktok merupakan platform yang berkembang pesat dengan basis pengguna yang sangat beragam, menawarkan konten yang beragam dan pengguna dapat berinteraksi secara realtime dengan konten satu sama lain sehingga menghasilkan data sentimen yang beragam. Analisis sentimen memungkinkan aplikasi TikTok untuk memahami opini dan perasaan

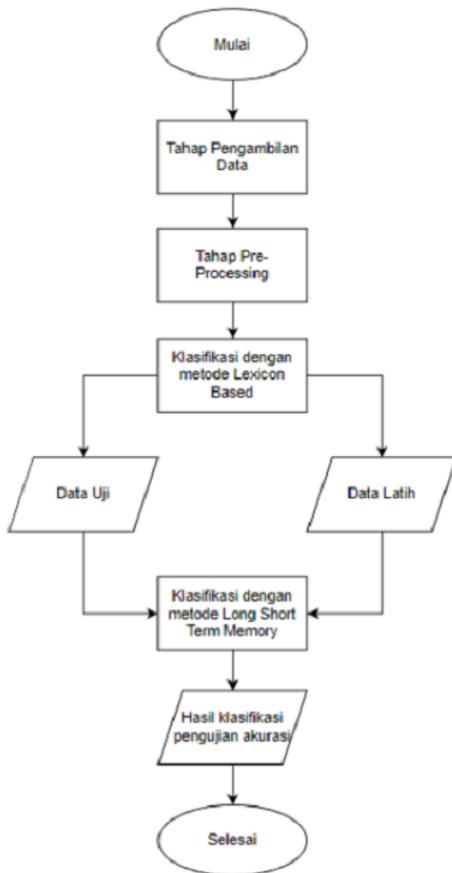
pengguna terhadap platform, konten, fitur dan pengalaman secara keseluruhan. Umpan balik ini dapat digunakan untuk meningkatkan kualitas pelayanan. Dengan memanfaatkan analisis sentimen secara efektif, aplikasi TikTok dapat memperkuat posisi sebagai platform terdepan di era digital yang dinamis. Kombinasi factor-faktor ini menjadikan aplikasi TikTok sebagai objek yang ideal untuk menerapkan analisis sentimen.

Penelitian ini bertujuan untuk melakukan klasifikasi kelas sentimen pada data yang berasal dari data ulasan aplikasi Tiktok dengan menerapkan algoritma Long Short-Term Memory (LSTM) dan metode Lexicon Based untuk mendapatkan pengelompokan positif, negatif dan netral. LSTM digunakan karena memiliki kemampuan menangkap pola dan konteks dalam teks dengan lebih baik. Metode Lexicon Based digunakan dalam penelitian ini karena metode ini memberikan kemudahan dalam interpretasi hasil klasifikasi, sederhana, layak dan praktis untuk analisis sentimen dari data media sosial, dengan kecesuaian dari data yang digunakan yaitu berupa opini pelanggan tentang suatu pelayanan jasa. Lexicon based didasarkan pada asumsi bahwa orientasi sentimen kontekstual

adalah jumlah dari orientasi sentimen setiap kata atau frase. Metode leksikon digunakan untuk mengekstrak sentimen dengan mengkombinasi lexical knowledge dan klasifikasi teks [4].

Metode LSTM adalah variasi dari Recurrent Neural Network (RNN) dengan kelebihan, diantaranya mampu menyimpan informasi jangka panjang, membaca, serta memperbarui informasi sebelumnya. Salah satu permasalahan yang ada pada RNN yang dapat diselesaikan dengan LSTM yaitu ketidakmampuan dalam menyimpan informasi ketika proses learning apabila terlalu banyaknya informasi yang harus disimpan [5].

Pada suatu penelitian metode LSTM digunakan untuk mengklasifikasikan review novel berbahasa Indonesia dengan menghasilkan 3 kelas yaitu sentimen positif, netral dan negatif. Diharapkan Metode LSTM memiliki hasil nilai akurasi yang lebih baik daripada metode naïve bayes yang memiliki nilai akurasi sebanyak 72.85%, precision 73%, recall 72%, dan f-measure 72% dibandingkan dengan hasil nilai akurasi Naïve Bayes yaitu dengan nilai akurasinya sebanyak 67.88%, precision 69%, recall 68%, dan f-measure 68% [6].



Gambar 1: Tahapan Analisis Sentimen

Penelitian lain metode LSTM digunakan pada ulasan aplikasi Tokopedia di Google Playstore dan menghasilkan akurasi yang cukup baik pada parameter 400 jumlah neuron dan fungsi aktivasi sigmoid. Parameter terbaik selanjutnya akan diuji

pada 20% dari dataset dan menghasilkan akurasi sebanyak 93,32%, presisi 95,17% dan nilai recall 97,15% [7]. Berdasarkan uraian tersebut, metode LSTM menghasilkan nilai akurasi cukup baik, maka pada penelitian ini dilakukan implementasi metode LSTM dan Lexicon Based untuk klasifikasi sentimen pada ulasan pengguna aplikasi TikTok pada Google Playstore.

Metode Penelitian

Kegiatan analisis sentimen terhadap data ulasan terdiri dari proses-proses dan sub-proses yang saling berhubungan. Proses dimulai dari pengambilan data, pemrosesan data, klasifikasi sentimen, hingga pengujian akurasi sentiment, seperti pada Gambar 1.

Tahap Pengambilan Data

Tahap ini merupakan tahap awal dalam melakukan analisis sentimen, Pada tahap ini meliputi 2 proses, yaitu text crawling dan penyimpanan data. Text crawling adalah suatu proses pemindaian halaman internet untuk membuat indeks dari data teks yang dicari [8].

Proses text crawling pada penelitian ini menggunakan python library "google_play_scraper". Proses ini merupakan proses utama pada analisis sentiment dalam mendapatkan dataset ulasan berbentuk ekstensi .csv dari Google Playstore. Setelah melakukan pengambilan data, tahap selanjutnya melakukan penyimpanan data.

Tahap Text Pre-processing

Pada tahap ini file yang disimpan akan diproses dengan tujuan untuk menghasilkan dataset yang singkat, padat, dan jelas untuk dilakukan analisis sentimen pada tahap berikutnya. Pada tahap ini meliputi beberapa sub-proses, yaitu case-folding, filtering, tokenizing, normalization, stemming kata dasar, dan diakhiri dengan penyimpanan data yang merupakan luaran dari proses text pre-processing. Sebelum masuk ke tahap text processing, harus didefinisikan terlebih dahulu file yang akan diproses dalam tahapan text pre-processing yang bertujuan untuk meminimalisir kata yang tidak berarti dalam proses selanjutnya [9].

Pada penelitian ini ekstensi file yang digunakan adalah .csv. Proses Clean Data. Pada tahap ini akan dilakukan penghapusan elemen-elemen yang tidak dibutuhkan seperti simbol, angka, double space, karakter satuan, dan lainnya. Data ulasan akan diperiksa apakah mengandung simbol, angka, dan lainnya. Jika mengandung simbol, angka, dan lainnya tersebut akan dihapus.

Proses Case Folding. Pada tahap ini mengubah huruf pada data ulasan yang berawal huruf kapital menjadi huruf kecil. data ulasan akan

diperiksa apakah mengandung huruf kapital atau tidak. Jika mengandung huruf kapital maka huruf tersebut akan diubah menjadi huruf kecil [10].

Proses Tokenizing. Pada tahap ini melakukan pemotongan string masukan berdasarkan kata-kata yang menyusunnya atau dengan kata lain pemecahan kalimat menjadi kata [10]. Data hasil dari tahap Tokenizing ini akan digunakan kembali pada tahap text pre-processing selanjutnya.

Proses Filtering. Pada tahap ini akan dilakukan penghapusan pada kosakata yang tidak diperlukan akan diperiksa dan dieliminasi. Jika terdapat kata sambung, kata depan, kata ganti, atau kata yang tidak ada hubungannya dalam analisis sentimen, maka kata tersebut akan dihilangkan sehingga mengurangi noise pada teks yang akan dilakukan klasifikasi, dan dapat meningkatkan akurasi. Data pada tahap filtering akan digunakan kembali pada tahap text preprocessing selanjutnya.

Proses Normalization. Pada tahap ini dilakukan perubahan kata yang tidak sesuai dengan EYD. Tahap ini dibagi menjadi dua langkah, yaitu konversi kata singkatan, dan konversi kata baku [10]. Data hasil dari tahap normalization ini akan digunakan kembali pada tahap text pre-processing selanjutnya.

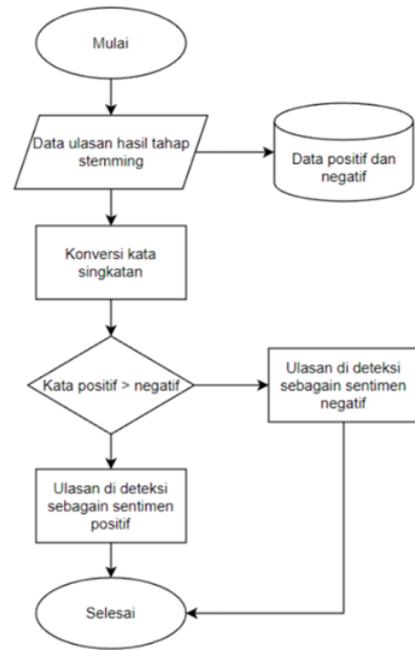
Proses Stemming. Pada tahap ini akan dilakukan perubahan kosakata berimbuhan yang terdapat dalam kalimat teks sumber akan diubah menjadi kata dasar dengan melakukan pemotongan terhadap imbuhan. Proses stemming kata dasar menggunakan algoritma yang dikembangkan oleh Nazief & Andiriani [11]. Data hasil dari tahap stemming ini merupakan tahap akhir dari text pre-processing selanjutnya simpan data yang telah melalui tahap text pre-processing.

Tahap Klasifikasi dengan Metode Lexicon Based

Metode Lexicon Based merupakan salah satu klasifikasi yang digunakan dalam penelitian ini. Gambar 2 adalah gambaran tahap Lexicon Based. Klasifikasi dengan metode lexicon adalah klasifikasi ulasan berdasarkan kata positif dan kata negatif pada hasil data ulasan yang telah dibersihkan pada tahap text pre-processing.

Hasil data yang telah melalui tahap preprocessing akan dibaca per-kata dan dicocokkan dengan kamus kata bersentimen, dengan hasil apabila suatu teks sumber memiliki jumlah kata bersentimen positif lebih banyak dari jumlah kata bersentimen negatif, maka teks sumber tersebut diklasifikasikan sebagai teks sumber bersentimen positif, begitu pula sebaliknya.

Data yang digunakan oleh peneliti terdiri dari 26083 record. Dataset ini terdiri dari 14 indeks hipertensi. Di bawah ini adalah rincian indikator atau atribut kumpulan data yang akan digunakan.



Gambar 2: *Klasifikasi Dengan Metode Lexicon Based*

Tahap Perancangan Long Short-Term Memory

Pada analisis sentiment, LSTM digunakan untuk memprediksi sentiment (positif, negative, netral) dari teks ulasan. Ulasan yang representasikan secara numerik akan dimasukkan ke dalam LSTM yang memproses urutan kata dan menangkap pola dan konteks. Selanjutnya pada setiap Langkah waktu, LSTM akan melakukan prediksi sentiment kata atau frasa berikutnya berdasarkan informasi yang telah diproses sebelumnya.

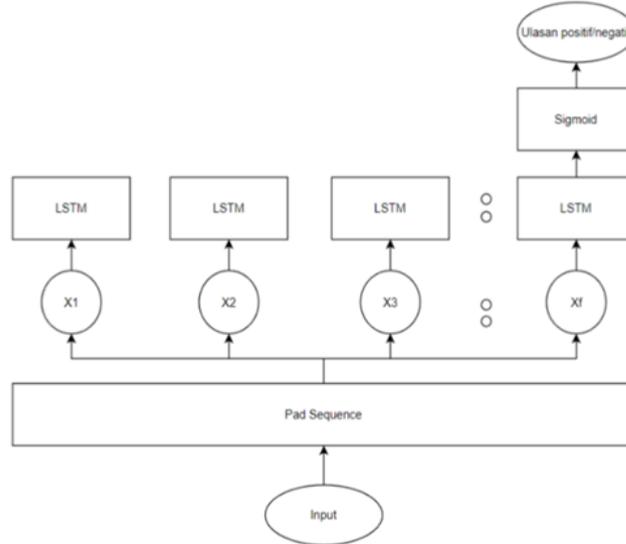
Algoritma LSTM memiliki memori yang cukup besar, memori jangka Panjang LSTM memungkinkan jaringan untuk mengingat informasi penting dari awal urutan data yang panjang dan menggunakannya untuk memproses elemen selanjutnya secara akurat, sehingga metode ini cocok untuk berbagai tugas yang melibatkan data sekuensial.

Tahapan awal dalam merancang sebuah model LSTM adalah melakukan import dataset hasil dari crawling data dan yang telah dilakukan proses pre-processing. Kemudian, pada dataset yang sudah di import dilakukan tokenizing yang dilakukan dengan cara yang berbeda menggunakan tokenization pada pre-processing. Proses-proses perubahan data yang ada pada dataset menjadi word index agar mudah dibaca dan diolah kedalam algoritma LSTM dan menghasilkan luaran berupa unique token.

Alur kerja dari algoritma LSTM digambarkan pada Gambar 3. Data di input terlebih dahulu sebelum masuk ke algoritma LSTM yaitu melalui tahap pad sequence/encode review. Selanjutnya, LSTM memproses hasil dari pad sequence. Setelah input pertama selesai diproses, maka setelah itu,

hasilnya akan dimasukkan kedalam memori, yang dimana hasil dari input sebelumnya akan digunakan untuk proses berikutnya yaitu untuk memeriksa input pertama dengan input kedua dan seterusnya. Hasil proses berikutnya dipengaruhi dari hasil input yang sebelumnya. Input berupa kalimat yang

terdiri dari beberapa kata. Pada proses embedding jumlah output embedding dimension adalah 16. Pada layer LSTM digunakan fungsi random untuk banyaknya jumlah output dari LSTM tersebut. Fungsi dropout digunakan untuk memutuskan koneksi agar model hasil pelatihan tidak overfitted.



Gambar 3: Alur Kerja Algoritma LSTM

```
#Scrape desired number of reviews
#Run kode ini jika ingin scrape data dengan jumlah tertentu. Ganti (misal, ingin scrape sejumlah 1000, maka ganti kode , count =

from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.ss.android.ugc.trill',
    lang='id', # defaults to 'en'
    country='id', # defaults to 'us'
    sort=Sort.MOST_RELEVANT, # defaults to Sort.MOST_RELEVANT you can use Sort.NEWEST to get newest reviews
    count=40000, # defaults to 100
    filter_score_with=None # defaults to None(means all score) Use 1 or 2 or 3 or 4 or 5 to select certain score
)
```

Gambar 4: Pengambilan Data (Text Crawling)

Tahap Pengujian

Pengujian model yang optimal dapat ditentukan berdasarkan parameter yang menghasilkan nilai terbaik.

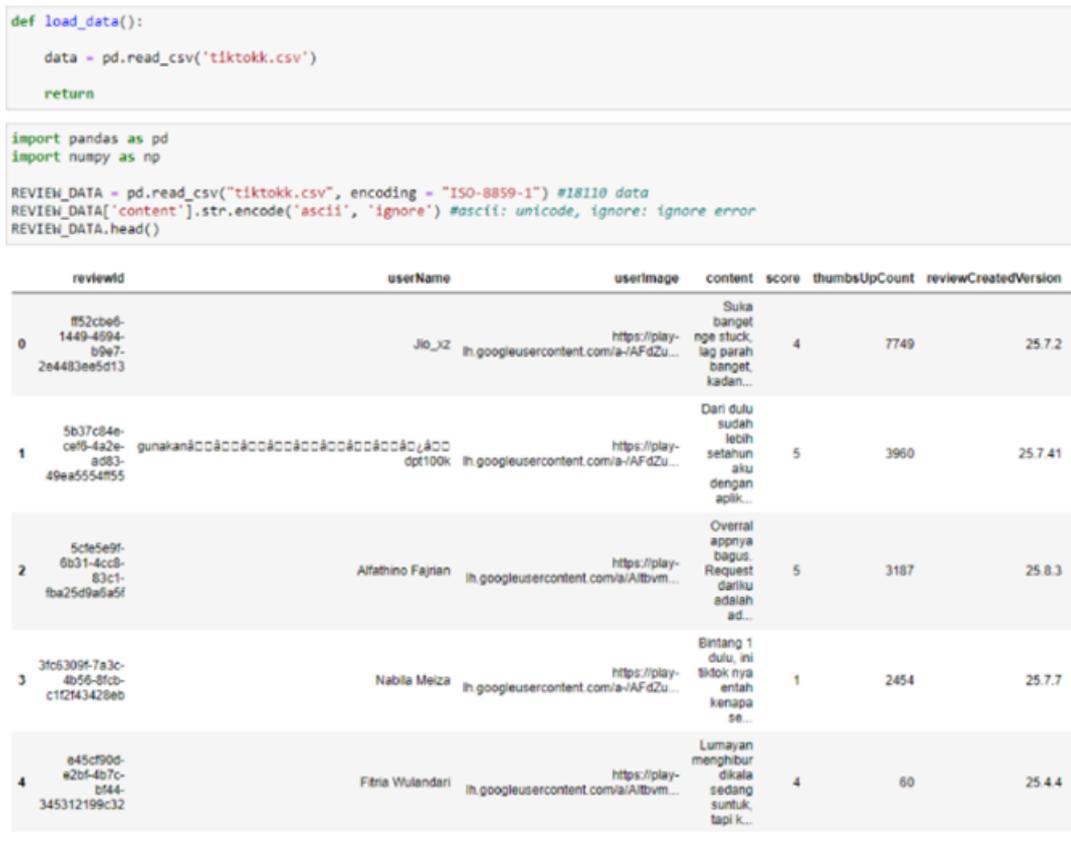
Pada penelitian ini penentuan model yang optimal dilakukan dengan pengujian dari beberapa parameter di antaranya adalah pengujian jumlah neuron dan pengujian fungsi aktivasi.

Hasil dan Pembahasan

Pengambilan data merupakan proses utama pada analisis sentimen untuk mendapatkan data ulasan pengguna, data yang diambil dari ulasan pengguna sebanyak 40.000 data. Data yang berhasil didapatkan dari proses pengambilan data (Text Crawling) yang dilakukan ditunjukkan pada Gambar 4. Setelah melakukan pengambilan data (Text Craw-

ling) ulasan TikTok pada Google Playstore, tahap selanjutnya melakukan penyimpanan data, penyimpanan data bertujuan untuk menyimpan data hasil dari Crawling Data kedalam komputer dan mengubah file menjadi berbentuk ekstensi .csv disimpan dengan nama tiktok.csv, file ini berguna untuk dapat diolah ketahap Text Pre-processing.

Text Pre-processing file yang disimpan akan diproses, dengan tujuan untuk menghasilkan data yang bersih, singkat, dan jelas untuk dilakukan proses ke tahap berikutnya pada analisis sentimen. Tahap ini memiliki beberapa sub-proses, yaitu case folding, filtering, tokenizing, normalization, stemming, dan diakhiri dengan penyimpanan data yang merupakan output dari proses text pre-processing. Data yang berhasil didapatkan dari proses text pre-processing yang dilakukan ditunjukkan pada Gambar 5. Contoh proses text pre-processing ditunjukkan pada Table 1.



Gambar 5: Proses Text Pre-processing

Tabel 1: Contoh Proses Text Pre-Processing

Data Ulasan	Tahap Clean Data	Tahap Case Folding	Tahap Tokenizing	Tahap Filtering	Tahap Normalizati on	Tahap Stemming
Kenapa setiap hasil rekam gerakan mulut gak sesuai dengan music ?, Padahal waktu ngerekam sesuai ?, saya udah update 2 kali, saya kira yang ke 2 bakal normal, eh ternyata tidak, mohon perbaikannya segera !!!!!!!!	Kenapa setiap hasil rekam gerakan mulut gak sesuai dengan musik ?, Padahal waktu ngerekam sesuai saya udah update kali saya kira yang ke bakal normal eh mohon perbaikannya segera	kenapa setiap hasil rekam gerakan mulut gak sesuai dengan musik padahal waktu ngerekam sesuai saya udah update kali saya kira yang ke bakal normal eh ternyata tidak mohon perbaikannya segera	['kenapa', 'setiap', 'hasil', 'rekam', 'gerakan', 'mulut', 'gak', 'sesuai', 'musik', 'ngerekam', 'mulut', 'gak', 'sesuai', 'update', 'kali', 'dengan', 'normal', 'eh', 'saya', 'kira', 'yang', 'ke', 'bakal', 'normal', 'eh', 'mohon', 'perbaikannya', 'segera']	['hasil', 'rekam', 'gerakan', 'mulut', 'gak', 'sesuai', 'musik', 'ngerekam', 'mulut', 'gak', 'sesuai', 'udah', 'update', 'kali', 'normal', 'eh', 'musik', 'mohon', 'rekam', 'sesuai', 'perbaikannya', 'segera']	['hasil', 'rekam', 'gerakan', 'mulut', 'rekam', 'mulut', 'tidak', 'sesuai', 'musik', 'rekam', 'sesuai', 'sudah', 'update', 'kali', 'normal', 'mohon', 'perbaikann ya', 'segera']	['hasil', 'rekam', 'gerak', 'mulut', 'tidak', 'sesuai', 'musik', 'rekam', 'sesuai', 'sudah', 'update', 'kali', 'normal', 'mohon', 'baik', 'segera']

Dataset yang digunakan pada penelitian ini terdiri dari 13.000 ulasan positif, 13.000 ulasan negative dan 14.000 ulasan netral. Pada penelitian

ini penentuan model yang optimal dilakukan dengan pengujian dari beberapa parameter di antaranya adalah pengujian jumlah neuron dan pen-

gujian fungsi aktivasi. Arsitektur jaringan dibentuk untuk menghasilkan akurasi yang optimal. Selanjutnya akan dilakukan penelitian model LSTM. Lapisan LSTM menggunakan jumlah neuron 16. Semua jaringan dihubungkan kepada neuron dengan lapisan fully connected dengan jumlah unit 3 berdasarkan banyaknya kelas pada penelitian ini.

Tahap terakhir adalah menggunakan fungsi aktivasi sigmoid. Setelah dibangun model, lalu dikonfigurasi terlebih dahulu dengan menggunakan optimasi Adam dan sparse categorical crossentropy sebagai parameter nilai loss dari model yang sudah terbentuk. Parameter lainnya untuk membantu proses training adalah ukuran batch size dan epoch. Berdasarkan urutan parameter yang digunakan, Gambar 6 adalah arsitektur jaringan pada model yang telah terbentuk.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 96, 32)	160000
lstm_1 (LSTM)	(None, 16)	3136
dropout_1 (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 3)	51

 Total params: 163,187
 Trainable params: 163,187
 Non-trainable params: 0

Gambar 6: Total Parameter Penelitian

Pengujian model yang optimal dapat ditentukan berdasarkan parameter yang menghasilkan nilai terbaik. Pada penelitian ini penentuan model yang optimal dilakukan dengan pengujian dari beberapa parameter di antaranya adalah pengujian jumlah neuron dan pengujian fungsi aktivasi.

Pengujian jumlah neuron, pengujian di mulai dengan menguji jumlah neuron pada lapisan. Dalam penelitian ini dilakukan pengujian pada jumlah lapisan 16, semakin banyak neuron yang dipakai akan semakin tinggi nilai dari test accuracy, semakin rendah nilai test loss dan semakin panjang waktu yang dipakai untuk memproses namun tidak signifikan. Hasil dari pada pengujian jumlah neuron menunjukan pada jumlah neuron sebesar 16 memberikan hasil test akurasi sebesar 89,89% dan diiringi dengan nilai test loss sebesar 0.53. Hasil pengujian jumlah neuron dapat dilihat pada Tabel 2.

Tabel 2: Hasil Pengujian Data Neuron

Jumlah Neuron	Embedding Dimension	Test Accuracy	Test Loss
16	32	89,89%	0.53

Setelah pengujian jumlah neuron dilakukan, selanjutnya dilakukan pengujian terhadap fungsi aktivasi menggunakan parameter yang telah digunakan sebelumnya yaitu menggunakan jumlah neu-

ron sebesar 16, dan embedding dimension sebesar 32. Pengujian ini dilakukan menggunakan sigmoid sebagai fungsi aktivasi. Sigmoid adalah gerbang yang mengatur seberapa banyak informasi yang lewat [12] dan memiliki akurasi sebesar 89,89%. Hasil pengujian aktivasi pada lapisan output dapat dilihat pada Tabel 3.

Tabel 3: Hasil Pengujian Fungsi Aktivasi

Jumlah Neuron	Embedding Dimension	Aktivasi	Test Accuracy	Test Loss
16	32	Sigmoid	89,89%	0.53

Berdasarkan pengujian parameter yang telah diuji, didapatkan model dengan nilai parameter yang optimal dengan jumlah neuron sebanyak 16, embedding dimension sebanyak 32 dan menggunakan fungsi aktivasi sigmoid. Kemudian parameter ini selanjutnya dilakukan training dan validasi. Proses training menggunakan epoch sebanyak 10 dan batch size sebanyak 128. Epoch menandakan satu siklus algoritma machine learning melakukan learning dari seluruh set data training [13].

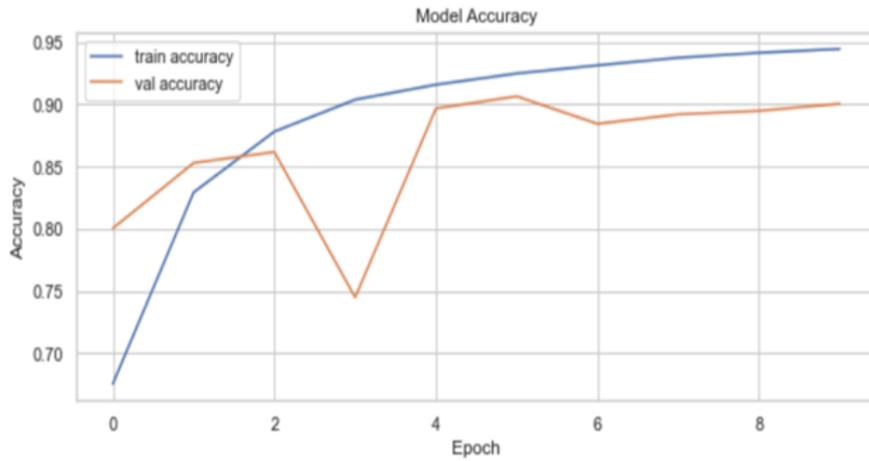
Dengan parameter yang sudah ditentukan akan terlihat beberapa akurasi dari data training dan validasi serta melihat nilai loss terendah. Model akan menyimpan epoch yang optimal pada nilai loss yang terendah selama proses epoch berlangsung. Hasil training yang dapat dilihat pada Gambar 7 dan Gambar 8.

Berdasarkan pada Gambar 7 untuk garis berwarna orange menunjukkan frekuensi nilai akurasi yang didapatkan dari epoch yang dilakukan sebanyak 10 kali dan didapatkan nilai akurasi sebesar 89,89% dan Gambar 8 untuk garis berwarna orange menunjukkan frekuensi nilai loss yang didapatkan dari epoch yang dilakukan sebanyak 10 kali dan didapatkan nilai akurasi sebesar 0,5365.

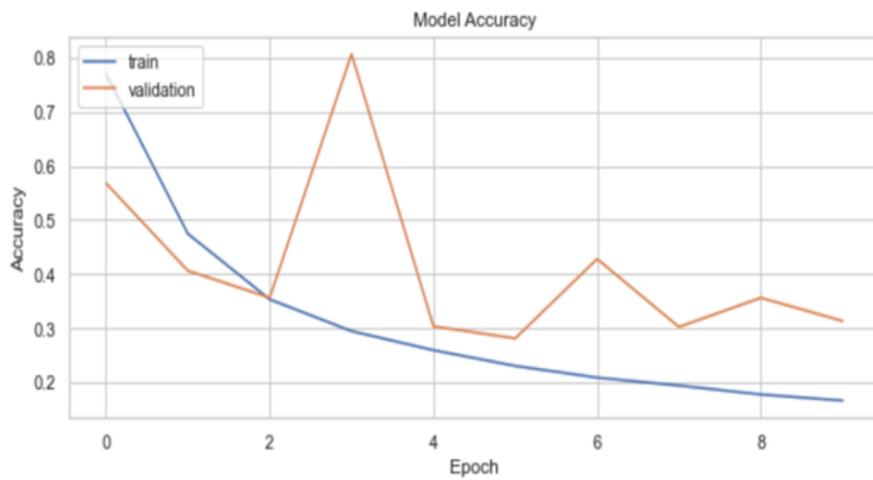
Confusion matrix digunakan untuk melakukan pengukuran guna mendapatkan nilai-nilai yang digunakan untuk pengukuran. Pada pengujian ini menunjukkan true positive sebanyak 2135 kalimat teridentifikasi sebagai ulasan positif, true negative sebanyak 4582 kalimat teridentifikasi sebagai kalimat negatif, false positive sebanyak 182 kalimat positif yang teridentifikasi sebagai ulasan negatif, false negative sebanyak 58 kalimat negatif yang teridentifikasi sebagai ulasan positif. Dengan hasil tersebut menghasilkan akurasi sebesar 90,05% confusion matrix dapat dilihat pada Gambar 9.

Pengujian Precision

Precision adalah nilai yang membandingkan antara jumlah ulasan positif yang teridentifikasi benar dengan jumlah kalimat yang dikategorikan ulasan positif dengan rumus 1 [14].

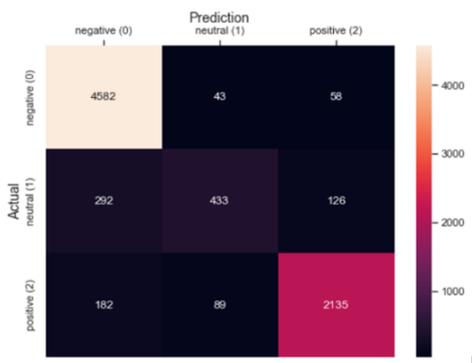


Gambar 7: Visualisasi Nilai Akurasi Training Dan Validasi



Gambar 8: Visualisasi Nilai Loss Training Dan Validasi

249/249 [-----] - 2s 6ms/step
Model Accuracy on Test Data: 0.9005037783375315



Gambar 9: Visualisasi Hasil Confusion Matrix

Berdasarkan perhitungan didapatkan nilai precision untuk mendeteksi ulasan positif adalah sebesar 92,14%. Angka tersebut menunjukkan hasil identifikasi cukup tepat dalam mengidentifikasi ulasan tersebut merupakan ulasan positif.

$$Precision = \frac{tp}{tp + fp} \quad (1)$$

$$Precision = \frac{2.135}{2.135 + 182} = 92,14\%$$

Pengujian Recall

Recall adalah nilai yang membandingkan antara jumlah ulasan positif yang teridentifikasi benar dengan jumlah hasil klasifikasi ulasan positif dengan rumus 2 [14].

Berdasarkan perhitungan didapatkan nilai recall untuk mendeteksi ulasan positif adalah sebesar 97,35%. Hasil tersebut menunjukkan dari 2.239 kalimat positif, 58 diantaranya adalah kalimat positif yang awalnya diberi label kalimat negatif.

$$Recall = \frac{tp}{fn + tp} \quad (2)$$

$$Recall = \frac{2.135}{58 + 2.135} = 97,35\%$$

Pengujian F1 Score

F1-score adalah nilai yang membandingkan rata-rata nilai pengujian precision dan pengujian recall yang dibobotkan dengan rumus 3 [15]. Berdasarkan perhitungan didapatkan nilai F1-Score sebesar

98,66%. Angka tersebut menunjukkan hasil bahwa model klasifikasi precision dan recall yang baik

$$F1Score = 2 * \frac{recall * precision}{recall + precision} \quad (3)$$

$$F1Score = 2 * \frac{2,14}{92,14} = 98,66\%$$

Penutup

Proses klasifikasi menggunakan metode Lexicon Based untuk menentukan kalimat yang memiliki makna sentimen positif, negatif, atau netral dan metode Long Short-Term Memory (LSTM) telah dilakukan, dan berdasarkan hasil pengujian diperoleh nilai akurasi yang didapatkan sebesar 90,05%, precision sebesar 92,14%, recall sebesar 97,35%, dan F-1 Score sebesar 98,66%, ulasan positif sebesar 30,0%, ulasan negatif sebesar 59,5%, dan ulasan netral sebesar 10,5%. Hasil analisis sentimen pada penelitian ini menunjukkan bahwa aplikasi TikTok memiliki kecenderungan sentimen negatif pada saat data ulasan diambil. Pengembangan dapat dilakukan dengan mengimplementasikan pada sebuah aplikasi yang memvisualisasikan hasil klasifikasi secara real-time berbasis website.

Daftar Pustaka

- [1] A. Mounika and S. Saraswathi, "Classification Of Book Reviews Based On Statement Analysis: A Survey", *Int. J. Res. Anal.*, vol. 6, pp. 150–155, 2019.
- [2] P. S. M. Suryani, L. Linawati dan K. O. Saputra, "Penggunaan Metode Naïve Bayes Classifier pada Analisis Sentimen Facebook Berbahasa Indonesia", *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 1, p. 145, doi: 10.24843/mite.2019.v18i01.p22, 2019.
- [3] D. A. Chontina Siahaan dan Abdiel Putra Laia, "Studi Literatur: Media Sosial 'Tiktok' Dan Pembentukan Karakter Remaja", *Syntax Lit. J. Ilm. Indones.*, vol. 7, no. 4, pp. 4939–4950, 2022.
- [4] A. Faadilah, "Analisis Sentimen Pada Ulasan Aplikasi Tokopedia di Google Play Store Menggunakan Metode Long Short Term Memory", *Skripsi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta*, pp. 1–46, 2020.
- [5] M. A. Nurrohmat and A. SN, "Sentiment Analysis of Novel Review Using Long Short-Term Memory Method", *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 3, p. 209, doi: 10.22146/ijccs.41236, 2019.
- [6] 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.
- [7] R. Cartwright, "Book Reviews: Book Reviews", *Perspect. Public Health*, vol. 130, no. 5, pp. 239–239, doi: 10.1177/1757913910379198, 2010.
- [8] M. A. Furqon, D. Hermansyah, R. Sari, A. Sukma, Y. Akbar dan N. A. Rakhmawati, "Analisis Jenis Posting Media Sosial Pemerintah Daerah Di Indonesia Berdasarkan Like Dan Analisis Sentimental Masyarakat", *J. Socioteknologi*, vol. 17, no. 2, p. 177, doi: 10.5614/sostek.itbj.2018.17.2.1, 2018.
- [9] R. A. S. Nurillah, M. Imrona dan A. Alamsyah, "Prediksi Pola Penyebaran Penyakit DBD di Kota Pagar Alam Menggunakan Long Short Term Memory (LSTM)", *eProceedings of Engineering*, vol. 8, no. 1, pp. 867–882, 2021.
- [10] M. S. Wibawa, "Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan", *J. Sist. dan Inform.*, vol. 11, no. December, pp. 167–174, doi: 10.13140/RG.2.2.21139.94241, 2017.
- [11] A. T. J. Harjanta, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining", *Jurnal UPGRIS*, Vol.1, No. 1, pp. 1–9, DOI: <https://doi.org/10.26877/jiu.v1i1%20Juni.804>, 2015.
- [12] W. W. Ariestya, I. Astuti and I. M. Wiryana, "Preprocessing for crawler of short message social media", *Proc. 3rd Int. Conf. Informatics Comput. ICIC 2018*, pp. 1–6, doi: 10.1109/IAC.2018.8780451, 2018.
- [13] B. Nazief and M. Adriani, "Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia", *Publikasi Internal, University of Indonesia, Depok*, 1996.
- [14] K. S. Nugroho, "Confusion matrix untuk evaluasi model pada supervised learning, Contoh: Untuk Pemodelan Klasifikasi Biner", *Medium*, 2019. diakses daring pada: <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-supervised-machine-learning-bc4b1ae9ae3f>.
- [15] N. Wanaspati, S. Anraeni and H. Darwis, "Support Vector Machine untuk Analisis Sentimen Masyarakat Terhadap Penggunaan Antibiotik di Indonesia", *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2196–2206, doi: 10.33022/ijcs.v12i4.3320, 2023.