

Perbandingan Algoritma LSTM dan BiLSTM Untuk Analisis Sentimen Multi-Class Media Sosial Twitter

Asnawiyah¹, Ricky Eka Putra²

^{1,3}Teknik Informatika, Universitas Negeri Surabaya

asnawiyah.20022@mhs.unesa.ac.id

rickyeka@unesa.ac.id

Abstrak— Perkembangan teknologi khususnya internet dan media sosial banyak digunakan oleh masyarakat. Salah satunya media sosial twitter sangat populer di Indonesia. Pengguna twitter dapat berinteraksi dan berbagi pendapat maupun opini melalui tweet. tweet ini berisi informasi berupa teks yang mengandung emosi pengguna. Analisis sentimen merupakan proses untuk menentukan emosi dari opini seseorang berupa teks. Penelitian ini, melakukan analisis multiclass menggunakan model long Short-Term Memory (LSTM) dan Bidirectional Long Short-Term Memory (BiLSTM). Model ini dipilih karena memiliki kemampuan dalam memahami data teks dan menangkap konteks temporal dalam kalimat. Penelitian dilakukan pada data sentiment yang berlabel yang diklasifikasikan dalam 5 kelas yaitu anger, fear happy, sadness, dan love. LSTM dan BiLSTM dilatih menggunakan representasi vector kata dan parameter yang dioptimalkan untuk meningkatkan akurasi. Proses implementasi juga membandingkan dengan menggunakan hyperparameter learning rate, batch size, epoch embedding glove, dan penyeimbang smote. Hasil evaluasi analisis sentiment multi-class pada media sosial twitter menunjukkan bahwa model BiLSTM memiliki akurasi 60% lebih baik dibandingkan LSTM memiliki akurasi 58%.

Kata Kunci— Analisis sentimen, multi-class, Twitter, LSTM, BiLSTM

I. PENDAHULUAN

Seiring perkembangan zaman diiringi berbagai inovasi dan kreativitas penemuan teknologi yang diciptakan oleh generasi muda. Saat ini, teknologi sangat melekat dengan manusia. Bahkan hampir aktifitas dirumah, dikantor, maupun diluar rumah membutuhkan teknologi untuk berkomunikasi. Banyaknya media komunikasi dan media sosial membutuhkan adanya jaringan internet. Berdasarkan laporan yang di catat oleh we are social mengenai pengguna akses internet di Indonesia menunjukkan mencapai 213 juta orang terhitung pada awal tahun 2023. Dapat disimpulkan jumlah 213 juta orang dari seluruh jumlah penduduk Indonesia 276,4 juta orang yaitu 77% pengguna internet. Dimana pengguna internet pada awal tahun 2022 berjumlah 202 juta orang sehingga mengalami kenaikan sebesar 5,44% di awal tahun 2023. Laporan we are social mencatat tahun yang mengalami pertumbuhan internet paling pesat di awal tahun 2016 dengan kenaikan sebesar 50,16%. Selain itu juga mencatat, rata – rata pengguna dalam menggunakan internet kisaran 7 jam 42 dalam kurun waktu 24 jam[1].

Pertumbuhan pengguna internet di Indonesia dipengaruhi oleh adanya smartphone. Perangkat smartphone berbasis

mobile ini banyak digunakan oleh pengguna karena memiliki kelebihan bentuknya simple sehingga mudah untuk dibawa dan mudah untuk mengakses media informasi dan hiburan. Salah satu media menyebar informasi yang membutuhkan akses internet yaitu aplikasi twitter [2]

Twitter merupakan media sosial yang sangat populer di Indonesia yang berdasarkan data salah satu pengguna twitter teraktif di dunia. Pengguna twitter dapat berinteraksi dan berbagi pendapat maupun opini melalui tweet. sesuai dengan negara Indonesia yang demokrasi dapat menyampaikan opini. Melalui tweet dapat mewakili berbagai opini keresahan dari Masyarakat. Opini yang disampaikan bisa tentang sebuah produk, layanan, toko publik, isu politik maupun film bioskop [3].

Dalam kehidupan sehari – hari banyak isu – isu yang trending sehingga dikomentari Masyarakat dalam media sosial twitter. Penelitian ini bertujuan untuk mengetahui pandangan Masyarakat yang dituliskan di media sosial twitter. Pandangan atau opini Masyarakat dapat berupa komentar positif ataupun negatif pada aplikasi X. menggunakan analisis sentimen ini dapat digunakan untuk mengidentifikasi dan menganalisis ekspresi emosional.

Analisis sentimen adalah salah satu cara untuk pengolahan data tekstual, mengekstraksi, dan mengolah informasi. Hasil analisis sentimen dapat diketahui opini positif dan negatif maupun emotional. Proses analisis sentimen membantu dalam memahami dan memperoleh informasi yang relevan dan tekstual dalam mengatasi opini pengguna twitter[4]

Terdapat penelitian terdahulu yang dilakukan oleh Nurkhairi Amini yang berjudul “Analisis Sentimen Dengan Polaritas Multi-Class Menggunakan Algoritma Klasifikasi *Support Vector Machine* dengan *Swarm Optimizer*”. Penelitian ini menggunakan dataset yang sama terdiri dari 4401 data dengan 8 label. Didapatkan akurasi 63,7% menggunakan SVM. Sedangkan Ketika menggunakan optimizer swarm didapatkan akurasi 65.8%[5].

Berdasarkan penelitian terdahulu, melakukan analisis sentimen membandingkan performa algoritma LSTM dan algoritma BiLSTM menggunakan parameter mengetahui sentimen positif dan negatif kemudian dilabeli ke dalam 8 kelas. Berdasarkan alasan diatas, penelitian ini dilakukan untuk menganalisis opini atau pendapat Masyarakat pada twitter menggunakan metode (LSTM) dan (BiLSTM) dengan membandingkan beberapa *hyperparameter*, penyeimbang smote dan *embedding glove*.

II. METODE



Gambar 1 Tahapan Penelitian

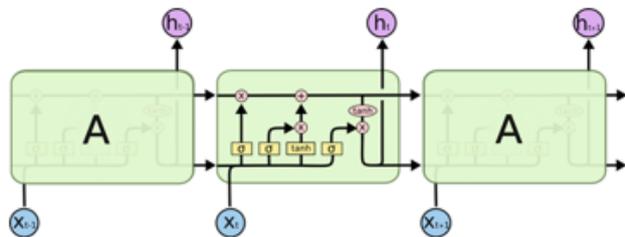
System klasifikasi yang dibangun dalam penelitian ini adalah yang dapat mengklasifikasikan sentiment publik pada media social twitter dalam bentuk ulasan Bahasa Indonesia. Pada penelitian ini, analisis sentiment digunakan untuk multi-class yaitu terdapat 5 class. Berdasarkan klasifikasi tweet maka perlu dikembangkan dengan membandingkan 2 algoritma yaitu algoritma LSTM dan BiLSTM.

A. Dataset

Penelitian ini menggunakan dataset yang diambil dari *kaggle*. Dataset ini berisi ulasan pengguna twitter yang sudah terlabeli menjadi 5 kelas. Dataset terdiri dari 4401 data dengan label marah (anger), senang (happy), takut (fear), sedih (sadness), dan cinta (love).

B. Metode

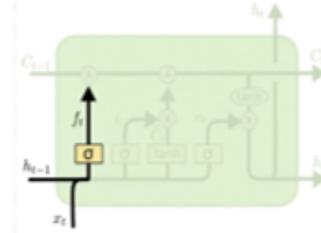
Tahap ini menerapkan model LSTM (Long Short-Term Memory) dan BiLSTM (Bidirectional Long Short-Term Memory) pada data. Berikut struktur jaringan LSTM:



Gambar 2 Jaringan LSTM

Pada struktur jaringan LSTM memiliki kondisi pada cell yaitu *cell state* dan *hidden state*. Struktur LSTM yaitu simpul

– simpul pada lapisan tersembunyi (*hidden layer*) digantikan oleh sel LSTM yang berfungsi untuk menyimpan informasi. Dalam sel LSTM terdapat tiga gerbang penting yaitu *input gate*, *forget gate*, dan *output gate*.



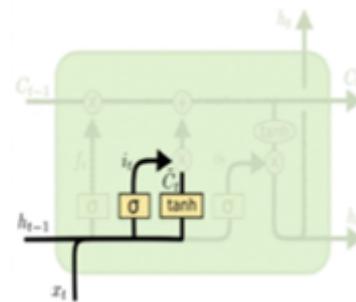
Gambar 3 Forget Gate

Forget Gate informasi yang ada pada input data yang akan diolah dan dipilah data yang layak untuk disimpan pada memory cells. Persamaan fungsi forget gate yaitu:

$$ft = \sigma (Wf [ht-1. xt] + bf) \quad \text{Persamaan 1}$$

Keterangan rumus:

- ft : forget gate
- xt : input variable x ke t
- σ : Fungsi Sigmoid
- Wf : Matriks bobot dari forget gate
- ht-1 : hidden state sebelumnya
- bf : Bias forget state



Gambar 4 Input Gate

Terdapat 2 gates yang dilakukan dalam input gates. Pertama menentukan nilai yang diperbarui dengan fungsi aktivasi sigmoid. Nilai aktivasi tanh akan membuat vector nilai terbaru yang akan di simpan di memory cell. Input gate layer Itu berguna untuk menentukan nilai yang akan diperbarui menggunakan fungsi aktivasi sigmoid dan tanh layer Ct berguna untuk membuat satu kandidat dengan nilai baru menggunakan fungsi aktivasi tanh.

Persamaan input gates

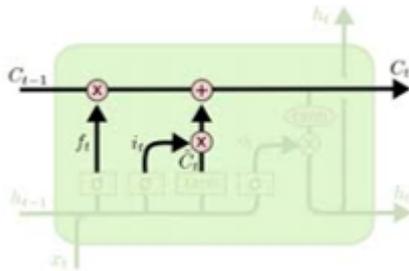
$$it = \sigma (Wi [ht-1. xt] + bi) \quad \text{Persamaan 2}$$

$$\dot{C}t = \tanh (W\dot{C}t[ht-1. xt] + bCt) \quad \text{Persamaan 3}$$

Keterangan persamaan:

- it : input gate
- $\dot{C}t$: cell aktivasi
- σ : Fungsi Sigmoid
- tan h : Fungsi aktivasi
- xt : input variable x ke t
- Wi : Bobot dari input gate

- $W\hat{C}_t$: Bobot dari cell aktivasi
- h_{t-1} : hidden state sebelumnya
- b_i : Bias input state
- $b\hat{C}_t$: Bias cell aktivasi



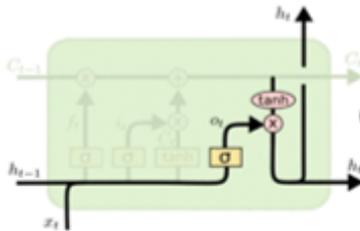
Gambar 5 Cell Gate

Cell gates akan berfungsi sebagai tempat pengganti memory cell yang lama dengan memory cell yang baru merupakan gabungan forgate gates dengan input gates. Pembaruan cell state terjadi dengan cara mengkalikan persamaan (1) dengan cell state sebelumnya kemudian ditambahkan dengan Langkah kedua yaitu persamaan (2) dan persamaan (3). Berikut persamaan dari cell gates:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad \text{Persamaan 4}$$

Keterangan persamaan:

- f_t : Forgate gate
- i_t : input gate
- C_t : cell gate
- \hat{C}_t : cell aktivasi
- C_{t-1} : cell gate sebelumnya



Gambar 6 Output Gate

Langkah terakhir yaitu output gates terdapat 2 langkah. Langkah pertama, lapisan sigmoid menggunakan bagian dari cell state yang akan menjadi keluaran. Kedua, keluaran dari cell state dimasukkan ke dalam tanh layer dan dikalikan dengan keluaran dari lapisan sigmoid.

$$\sigma = \sigma (W_o [h_{t-1}, x_t] + b_o) \quad \text{Persamaan 5}$$

$$h_t = o_t * \tanh (C_t) \quad \text{Persamaan 6}$$

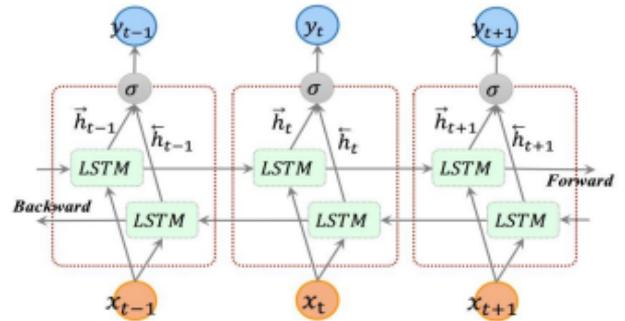
Keterangan persamaan:

- σ : Fungsi sigmoid
- W_o : Bobot dari output gate
- \tanh : Fungsi aktivasi
- h_{t-1} : hidden state sebelumnya
- x_t : input variable x ke t
- b_o : Bias output state

- C_t : Bobot dari cell gate

Tiga gerbang mengatur aliran informasi di dalam sel LSTM dan memungkinkan sel untuk mengontrol seberapa banyak informasi dari waktu sebelumnya untuk disimpan.

Penerapan metode BiLSTM (Bidirectional Long Short-Term Memory) pada data. Arsitektur Bi-LSTM mengacu pada gambar:



Gambar 7 Jaringan BiLSTM

Proses forward pada arsitektur BiLSTM dapat ditulis sebagai $\vec{h}_t = \text{LSTM}(x_t, h_{t-1})$, sedangkan pada proses backward LSTM ditulis sebagai $\overleftarrow{h}_t = \text{LSTM}(x_t, h_{t+1})$. Output akhir dari BiLSTM yaitu menggabungkan dari forward dan backward ditulis sebagai $h_t = [\vec{h}_t, \overleftarrow{h}_t]$

Keterangan:

\vec{h}_t = Nilai keluaran dari LSTM maju

\overleftarrow{h}_t = Nilai keluaran dari LSTM mundur

Perhitungan Forgate Gate

$f_t = \frac{1}{2} * (f_t \text{ forward} + f_t \text{ backward})$

III. PEMBAHASAN

Tahapan – tahapan yang dilakukan pada proses implemetasi sebagai berikut:

A. Preprocessing Data

Pada Langkah ini dilakukan preprocessing data untuk mengoptimalkan fitur data yang memiliki makna yang sama. Tahap pertama yaitu cleaning data untuk menghilangkan tanda baca, angka url pada sentiment. Hasil dari cleaning data dapat dilihat pada Tabel I.

Tabel I Cleaning Data

Sebelum	Sesudah
Ya Allah, hanya Engkau yang mengetahui rasa sakit dihati ini. Sembuhkanlah. Ya Allah	Ya Allah hanya Engkau yang mengetahui rasa sakit dihati ini Sembuhkanlah Ya Allah

Tahap kedua yaitu *case foalding* untuk mengubah sentiment yang memiliki huruf besar menjadi huruf kecil. Tujuannya agar menciptakan konsistensi dalam kata dan

mengurangi jumlah variasi sehingga kompleksitas data menjadi lebih kecil. Hasil dari *case foalding* dapat dilihat pada Tabel I.

Tabel II Case Foalding

Sebelum	Sesudah
Ya Allah hanya Engkau yang mengetahui rasa sakit dihati ini Sembuhkanlah Ya Allah	ya allah hanya engkau yang mengetahui rasa sakit dihati ini sembuhkanlah ya allah

Tahap ketiga yaitu *tokenization* untuk memisahkan teks menjadi sebuah token kata. *Tokenization* prosesnya menggunakan library *nlk.tokenize* pada python yaitu *word_tokenize*. Membagi teks menjadi token yang lebih kecil dapat mempermudah memproses dan menganalisis kata satu per satu. Hasil dari *tokenization* dapat dilihat pada Tabel III.

Tabel III Tokenization

Sebelum	Sesudah
ya allah hanya engkau yang mengetahui rasa sakit dihati ini sembuhkanlah ya allah	'ya', 'allah', 'hanya', 'engkau', 'yang', 'mengetahui', 'rasa', 'sakit', 'di', 'hati', 'ini', 'sembuhkanlah', 'ya', 'allah'

Tahap keempat yaitu normalisasi. Normalisasi adalah sebuah langkah diambil untuk mengubah singkatan, akronim, atau ambigu kata-kata dalam dokumen. Normalisasi dapat diatasi data yang tidak seimbang. kata-kata yang diperoleh peneliti dengan menganalisis dan memutarnya menjadi kata baku menurut KBBI. Hasil dari normalisasi dapat dilihat pada Tabel IV.

Tabel IV Normaisasi

Sebelum	Sesudah
Mlm	Malam
Kj	Kejadian
Tmn	Teman
Mntn	Menonton
Msk	Masuk
Plg	pulang

Tahap kelima yaitu *stopword removal*. Tahapan ini kata yang sering muncul dan tidak memiliki arti penting dan makna seperti kata depan dan penghubung akan dihilangkan. Proses *stopword removal* ini memanfaatkan library dari python yaitu *nlk.corpus*. Hasil dari *stopword removal* dapat dilihat pada Tabel V.

Tabel V Stopword Removal

Sebelum	Sesudah
'ya', 'allah', 'hanya', 'engkau', 'yang', 'mengetahui',	'allah', 'engkau', 'mengetahui', 'rasa', 'sakit', 'hati', 'semuhkanlah',

'rasa', 'sakit', 'di', 'hati', 'ini', 'sembuhkanlah', 'ya', 'allah'	'allah'
---	---------

Tahap keenam yaitu *steaming*. *Steaming* merupakan tahapan untuk menemukan bentuk utama dari sebuah kata menjadi kata dasar. Hasil dari *steaming* dapat dilihat pada Tabel VI.

Tabel VI Steaming

Sebelum	Sesudah
'allah', 'engkau', 'mengetahui', 'rasa', 'sakit', 'hati', 'sembuhkanlah', 'allah'	'allah', 'engkau', 'tahu', 'rasa', 'sakit', 'hati', 'sembuh', 'allah'

B. Perbandingan Hyperparameter

Berikut tabel perbandingan LSTM menggunakan hyperparameter batch size 8, 16, 32, 64 dan 128. Epoch menggunakan 5 dan learning rate 0.001

Metode LSTM

Tabel VII Hasil Akurasi Model LSTM dengan Membandingkann Hyperparameter Batch Size

Hyperparameter			
Epoch	Batch Size	Learning Rate	accuracy
5	8	0.001	0.5857
5	16	0.001	0.5743
5	32	0.001	0.5789
5	64	0.001	0.5732
5	128	0.001	0.5709

Metode Bilstm

Tabel VIII Hasil Akurasi Model BiLSTM dengan Membandingkann Hyperparameter Batch Size

Hyperparameter			
Epoch	Batch Size	Learning Rate	accuracy
5	8	0.001	0.5437
5	16	0.001	0.5834
5	32	0.001	0.5539
5	64	0.001	0.5868
5	128	0.001	0.5664

Berikut tabel perbandingan LSTM dan BiLSTM menggunakan hyperparameter learning rate 0.1, 0.01, 0.001. Epoch menggunakan 5 dan batch size 32

Metode LSTM

Tabel IX Hasil Akurasi Model LSTM dengan Membandingkann Hyperparameter Learning Rate

Hyperparameter			
Epoch	Batch Size	Learning Rate	accuracy
5	32	0.1	0.2270
5	32	0.01	0.5482
5	32	0.001	0.5721

5	32	0.001	0.5709
15	32	0.001	0.5403
25	32	0.001	0.5585

Berdasarkan tabel XII dan XIII nilai akurasi yang paling tinggi pada model LSTM dan BiLSTM ketika menggunakan nilai learning rate 0.001 dan batch size 32 yaitu dengan perpaduan epoch 5, 15, dan 25 nilai akurasi paling tinggi dengan hyperparameter epoch 5.

Metode BiLSTM

Tabel X Hasil Akurasi Model BiLSTM dengan Membandingkann Hyperparameter Learning Rate

Hyperparameter			
Epoch	Batch Size	Learning Rate	accuracy
5	32	0.1	0.2497
5	32	0.01	0.5335
5	32	0.001	0.5698

Berdasarkan tabel IX dan X nilai akurasi yang paling tinggi pada model LSTM dan BiLSTM ketika menggunakan Epoch menggunakan 5 dan batch size 32 yaitu dengan perpaduan nilai learning rate 0.1, 0.01, dan 0.001 didapatkan nilai akurasi paling tinggi pada learning rate 0.001. Dapat disimpulkan bahwa model LSTM dan BiLSTM memiliki akurasi paling tinggi menggunakan hyperparameter nilai learning rate 0.001.

Berikut tabel perbandingan LSTM menggunakan hyperparameter epoch 5, 15, 25. Batch size menggunakan 32 dan lerning rate 0.001

Metode LSTM

Tabel XI Hasil Akurasi Model LSTM dengan Membandingkann Hyperparameter Epoch

Hyperparameter			
Epoch	Batch Size	Learning Rate	accuracy
5	32	0.001	0.5607
15	32	0.001	0.5675
25	32	0.001	0.5607

Metode Bilstm

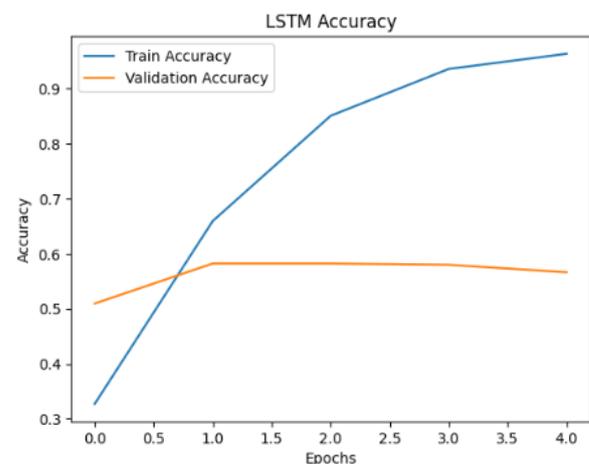
Tabel XII Hasil Akurasi Model BiLSTM dengan Membandingkann Hyperparameter Epoch

Hyperparameter			
Epoch	Batch Size	Learning Rate	accuracy

C. Model terbaik

Pada model ini menggunakan hyperparameter yang telah dilakukan percobaan dengan accuracy paling tinggi yaitu menggunakan batch size = 32, learning rate = 0.001 dan epoch = 5. Sehingga didapatkan grafik akurasi sebagai berikut:

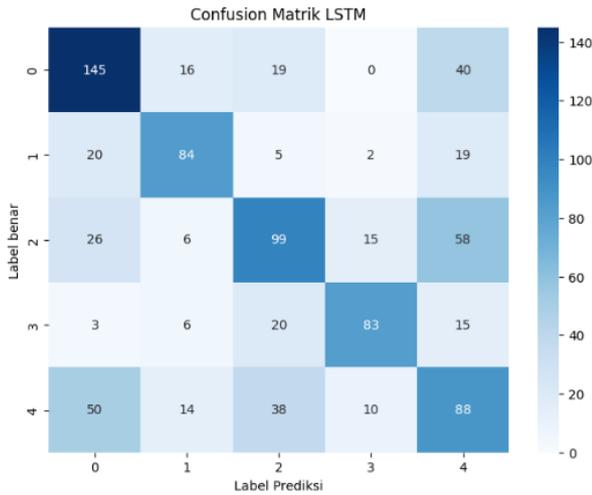
Metode LSTM



Gambar 8 Grafik Akurasi Model LSTM

Gambar 8 menunjukkan grafik akurasi model LSTM menunjukkan garis biru akurasi data train naik secara signifikan dari epoch pertama hingga epoch 3. Namun, pada saat mendekati 100% mulai melambat. Model ini baik dalam memprediksi data pelatihan.

Akurasi validation hanya mengalami sedikit peningkatan diawal epoch 1. Kemudian tetap stabil pada angka 0.5 – 0.6. dapat disimpulkan perbedaan besar antara akurasi train dan validation menunjukkan model mengalami overfitting sehingga kehilangan kemampuan untuk menggeneralisasi data baru.



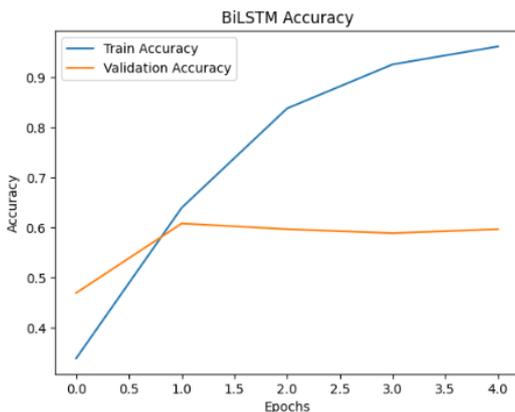
Gambar 9 Confusion Matriks Model LSTM

Gambar 4.27 diatas merupakan confusion matriks dari model LSTM untuk klasifikasi dengan 5 kelas dengan label 0 hingga 4. Elemen diluar diagonal utama menunjukkan jumlah kesalahan klasifikasi. Misalnya: kelas 0 salah klasifikasi sebagai kelas 1 sebanyak 16. Model ini memiliki performa yang cukup baik dalam mengenali kelas 0 dan kelas 2, tetapi kesulitan mengklasifikasikan pada kelas 4.

Perhitungan akurasi yang didapatkan dari confusion matriks

$$\begin{aligned}
 \text{Akurasi} &= \text{jumlah prediksi benar} / \text{jumlah total sampel} \\
 \text{Akurasi} &= 145 + 84 + 99 + 83 + 88 / 145 + 20 + 26 + 3 + 50 \\
 &+ 16 + 84 + 6 + 6 + 14 + 19 + 5 + 99 + 20 + 38 + 0 + 2 + 15 + \\
 &83 + 10 + 40 + 19 + 58 + 15 + 88 \\
 &= 499/881 \\
 &= 0.566 \\
 &= 0.57
 \end{aligned}$$

Metode BiLSTM

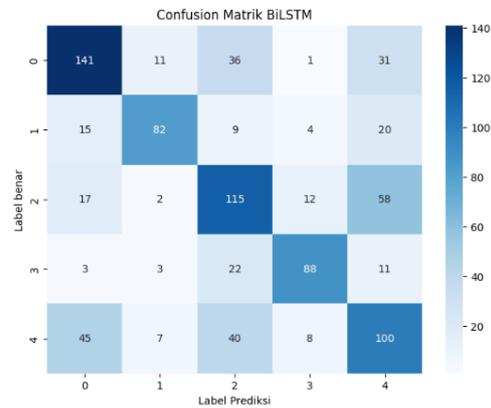


Gambar 10 Grafik Akurasi Model BiLSTM

Gambar 10 menunjukkan grafik akurasi model BiLSTM hampir sama dengan model LSTM menunjukkan garis biru akurasi data train naik secara signifikan dari epoch pertama

hingga epoch 3. Namun, pada saat mendekati 100% mulai melambat. Model ini baik dalam memprediksi data pelatihan.

Akurasi validation hanya mengalami sedikit peningkatan diawal epoch 1. Kemudian tetap stabil pada angka 0.5 – 0.6. dapat disimpulkan perbedaan besar antara akurasi train dan validation menunjukkan model mengalami overfitting sehingga kehilangan kemampuan untuk menggeneralisasi data baru.



Gambar 11 Confusion Matriks BiLSTM

Gambar 11 diatas merupakan confusion matriks dari model BiLSTM untuk klasifikasi dengan 5 kelas dengan label 0 hingga 4. Confusion matriks diatas dapat disimpulkan sebagai berikut:Model ini memiliki performa yang cukup baik dalam mengenali kelas 0 dan kelas 2, tetapi kesulitan mengklasifikasikan pada kelas 4.

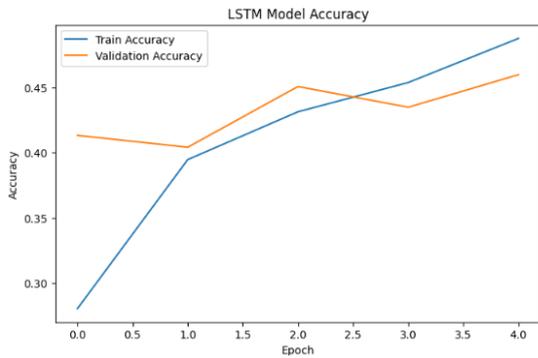
Perhitungan akurasi = jumlah prediksi benar / jumlah total sampel

$$\begin{aligned}
 \text{Akurasi} &= 141 + 82 + 115 + 88 + 100 / 141 + 15 + 17 + 3 + \\
 &45 + 11 + 82 + 2 + 3 + 7 + 36 + 9 + 115 + 22 + 40 + 1 + 4 + \\
 &12 + 88 + 8 + 31 + 20 + 58 + 11 + 100 \\
 &= 526/881 \\
 &= 0.597 \\
 &= 0.60
 \end{aligned}$$

D. Embedding Glove

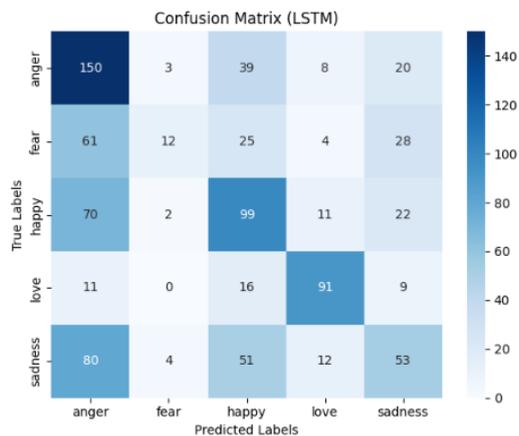
Glove adalah *embedding pre-trained* yang telah dipelajari dari dataset besar seperti Wikipedia. Percobaan ini menggunakan nilai hyperparameter terbaik yang telah didapatkan. Berikut hasil pemodelan menggunakan glove:

Metode LSTM



Gambar 12 Grafik Akurasi model LSTM Glove

Berdasarkan gambar 12 menunjukkan grafik model LSTM menggunakan feature creation glove akurasi pada data latih meningkat secara konsisten. Namun, akurasi pada data validasi mengalami peningkatan hingga epoch 3. Pada epoch 4 mengalami sedikit penurunan. Hal ini menandakan adanya overfitting, dimana model fokus pada data latih sehingga kehilangan kemampuan generalisasi pada data validasi.



Gambar 13 Confusion Matriks Model LSTM Glove

Berdasarkan tabel 13 model LSTM glove performa tiap kelas:

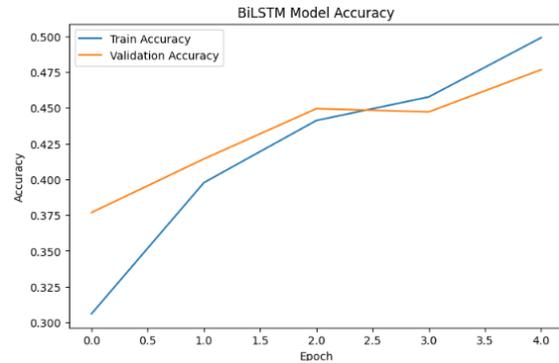
Kelas anger: model sangat baik dalam mengklasifikasikan emosi anger dan prediksi benar cukup tinggi, tetapi kadang salah memprediksi emosi sebagai kelas happy atau sadness
 Kelas fear: performa model pada kelas ini kurang optimal, karena banyak emosi fear yang salah diprediksi sebagai anger.
 Kelas happy: model cukup baik dalam mengenali happy, namun terdapat kesalahan cukup signifikan dalam kelas anger.
 Kelas love: model ini cukup baik dalam mengklasifikasikan love, meskipun ada kesalahan ke kelas happy dan kelas sadness.
 Kelas sadness: model ini sering salah memprediksi sadness sebagai anger, menunjukkan kesulitan dalam membedakan emosi ini.

Perhitungan akurasi

Akurasi = jumlah prediksi benar / jumlah total sampel

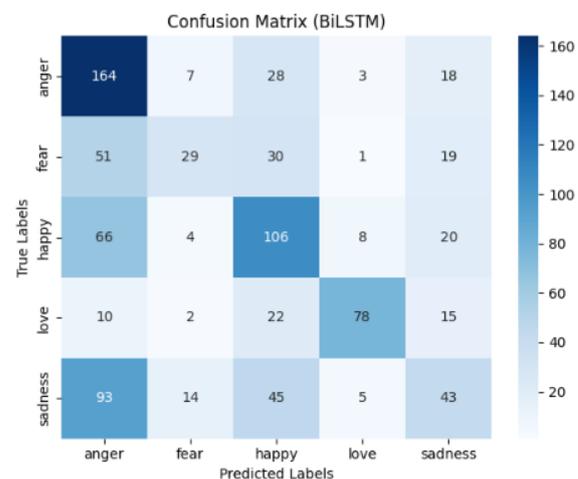
$$\begin{aligned}
 \text{Akurasi} &= 150 + 12 + 99 + 91 + 53 / 150 + 61 + 70 + 11 + \\
 &80 + 3 + 12 + 2 + 0 + 4 + 39 + 25 + 99 + 16 + 51 + 8 + 4 + 11 \\
 &+ 91 + 12 + 20 + 28 + 22 + 9 + 53 \\
 &= 405/881 \\
 &= 0.459 \\
 &= 0.46\text{BiLSTM}
 \end{aligned}$$

Metode BiLSTM



Gambar 14 Grafik Akurasi model BiLSTM Glove

Berdasarkan gambar 14 menunjukkan grafik dengan akurasi metode BiLSTM dengan feature creation glove pada data latih meningkat secara konsisten. Namun, akurasi pada data validasi mengalami peningkatan hingga epoch 3. Pada epoch 4 mengalami sedikit penurunan. Pada epoch 5 mengalami peningkatan lagi. Hal ini menandakan adanya overfitting pada data validasi, dimana model fokus pada data latih sehingga kehilangan kemampuan generalisasi pada data validasi.



Gambar 15 Confusion Matriks model BiLSTM Glove

Berdasarkan gambar 4.36 model BiLSTM glove performa tiap kelas:

Kelas anger: model sangat baik dalam mengklasifikasikan emosi anger dan prediksi benar cukup tinggi, tetapi kadang salah memprediksi emosi sebagai kelas happy atau sadness.
 Kelas fear: performa model pada kelas ini kurang optimal,

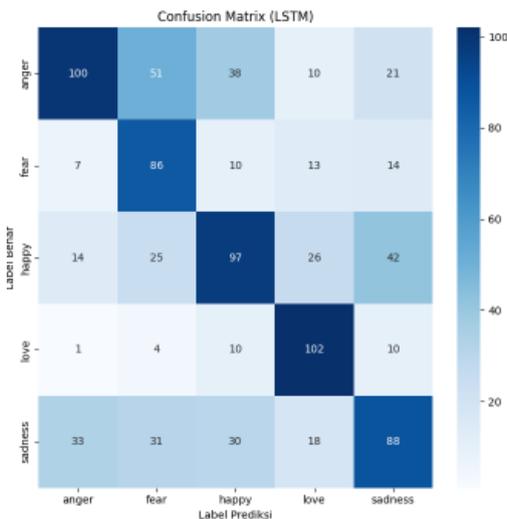
karena banyak emosi fear yang salah diprediksi sebagai anger. Kelas happy: model cukup baik dalam mengenali happy, namun terdapat kesalahan cukup signifikan dalam kelas anger. Kelas love: model ini cukup baik dalam mengklasifikasikan love, meskipun ada kesalahan ke kelas happy dan kelas sadness. Kelas sadness: model ini sering salah memprediksi sadness sebagai anger, menunjukkan kesulitan dalam membedakan emosi ini.

Akurasi = jumlah prediksi benar / jumlah total sampel

$$\begin{aligned} \text{Akurasi} &= 164 + 29 + 106 + 78 + 43 / 164 + 51 + 66 + 10 + \\ &93 + 7 + 29 + 4 + 2 + 14 + 28 + 30 + 106 + 22 + 45 + 3 + 1 + \\ &8 + 78 + 43 + 18 + 19 + 20 + 15 + 43 \\ &= 420/881 \\ &= 0.476 \\ &= 0.48 \end{aligned}$$

E. Perbandingan menggunakan Penyeimbang SMOTE

Pada perbandingan menggunakan penyeimbang SMOTE data ini membagi dataset 4401 dengan 5 kelas sehingga jumlahnya 881. Data dilakukan undersampling untuk data yang lebih dari 881 atau kelas mayoritas yaitu kelas anger dan kelas happy. Untuk kelas minoritas yaitu kelas sadness, kelas fear, kelas love dilakukan oversampling dengan ditambahkan data sintesis. Percobaan ini menggunakan hyperparameter terbaik yang telah didapatkan.



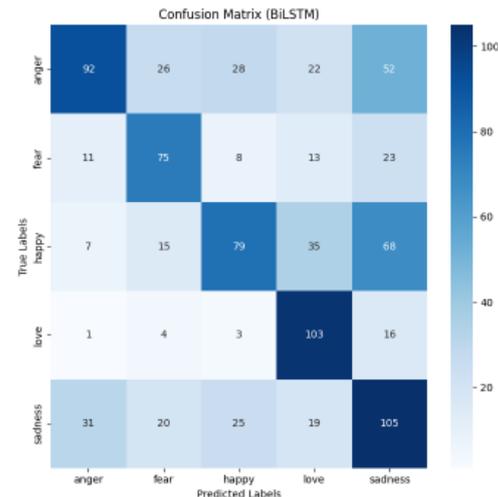
Gambar 16 Confusion Matriks Smote LSTM

Berdasarkan tabel 16 menunjukkan confusion matriks bahwa model ini mengalami kesulitan dalam membedakan emosi yang mirip seperti kelas marah (anger) dengan kelas takut (fear) dan kelas senang (happy) dengan kelas sedih (sadness). Dikarenakan kemiripan pola Bahasa yang digunakan dalam data untuk emosi tersebut. Kelas yang memiliki presiksi benar cukup baik yaitu kelas cita (love).

Perhitungan akurasi jumlah prediksi benar / jumlah total sampel

$$\begin{aligned} \text{Akurasi} &= 100 + 86 + 97 + 102 + 88 / 100 + 7 + 14 + 1 + 33 \\ &+ 51 + 86 + 25 + 4 + 31 + 36 + 10 + 97 + 10 + 30 + 10 + 13 + \\ &26 + 102 + 18 + 21 + 14 + 42 + 10 + 88 \\ &= 473/881 \\ &= 0.536 \\ &= 0.54 \end{aligned}$$

Metode BiLSTM



Gambar 17 Confusion Matrik Smote BiLSTM

Berdasarkan gambar 17 menunjukkan tabel confusion matriks bahwa model BiLSTM mengalami kesulitan dalam membedakan emosi yang mirip seperti kelas marah (anger) dengan kelas takut (fear) dan kelas senang (happy) dengan kelas sedih (sadness). Dikarenakan kemiripan pola Bahasa yang digunakan dalam data untuk emosi tersebut. Kelas yang memiliki presiksi benar cukup baik yaitu kelas sedih (sadness). Model LSTM setelah dilakukan normalisasi akurasinya menjadi 0.52. Perhitungan akurasi

$$\begin{aligned} \text{Akurasi} &= \text{jumlah prediksi benar} / \text{jumlah total sampel} \\ \text{Akurasi} &= 92 + 75 + 79 + 103 + 105 / 92 + 11 + 7 + 1 + 31 \\ &+ 26 + 75 + 15 + 4 + 20 + 28 + 8 + 79 + 3 + 25 + 22 + 13 + 35 \\ &+ 103 + 19 + 52 + 23 + 68 + 16 + 105 \\ &= 454/881 \\ &= 0.515 \\ &= 0.52 \end{aligned}$$

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan berjudul “Perbandingan Algoritma LSTM Dan BiLSTM Untuk Analisis Sentimen Multi Class Media Sosial Twitter” didapatkan kesimpulan sebagai berikut:

1. Untuk meningkatkan performa metode LSTM dan BiLSTM ada beberapa percobaan yaitu: Membandingkan hypeparameter batch size, learning rate,

- dan epoch sehingga didapatkan akurasi dari LSTM 0.57 dan BiLSTM 0.60 dengan menggunakan batch size 32, learning rate 0.001, dan epoch 5. 2. 2.
2. Membandingkan dengan embedding glove menggunakan hyperparameter terbaik sehingga didapatkan akurasi LSTM yaitu 0.46 dan BiLSTM yaitu 0.48.
 3. Membandingkan dengan penyeimbang smote menggunakan hyperparameter terbaik sehingga didapatkan akurasi LSTM yaitu 0.54 dan BiLSTM yaitu 0.52.

Berdasarkan percobaan untuk metode LSTM dan BiLSTM dengan membandingkan hyperparameter, embedding Glove dan penyeimbang smote bahwa yang mempengaruhi untuk mendapatkan performa yang baik ialah nilai learning rate.

V. SARAN

Berdasarkan penelitian yang telah dilakukan adapun saran yang dapat diberikan oleh penulis sebagai berikut:

1. Penelitian ini dapat nilai akurasinya dapat lebih dimaksimalkan lagi dengan menambahkan beberapa kombinasi percobaan pada nilai *dropout*, *activation* dan parameter lainnya.
2. Menggunakan algoritma klasifikasi *Deep Learning* lainnya seperti *Convolutional Neural Network* (CNN)

REFERENSI

- [1] Cindy Mutia Annur. 2023. "Pengguna Internet Di Indonesia Tembus 213 Juta Orang Hingga Awal 2023." Databoks.Katadata.Co.Id. 2023. <https://databoks.katadata.co.id/datapublish/2023/09/20/pengguna-internet-di-indonesia-tembus-213-juta-orang-hingga-awal-2023>.
- [2] Mahardhika, Sunggiale Vina, Iqlima Nurjannah, Isnaini Ila Ma'una, and Zahrotun Islamiyah. 2021. "Faktor-Faktor Penyebab Tingginya Minat Generasi Post-Millennial Di Indonesia Terhadap Penggunaan Aplikasi Tik-Tok." *Sosearch: Social Science Educational Research* 2 (1): 40–53.
- [3] Widowati, Tanthy Tawaqalia, and Mujiono Sadikin. 2020. "Analisis Sentimen Twitter Terhadap Tokoh Publik Dengan Algoritma Naive Bayes Dan Support Vector Machine." *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer* 11 (2): 626–36.
- [4] Puad, Salim, Garo Garo, and Agung Susilo Yuda Irawan. 2023. "Analisis Sentimen Masyarakat Pada Twitter Terhadap Pemilihan Umum 2024 Menggunakan Algoritma Naive Bayes." *JATI (Jurnal Mahasiswa Teknik Informatika)* 7 (3): 1560–66.
- [5] Rozy, Nurul Faizah, Nurkhairi Amini, Nashrul Hakiem, and Sandra Hakiem Afrizal. 2023. "Analysis of Multi-Class Sentiment on Indonesian Twitter Using Support Vector Machine Classification Algorithm with Particle Swarm Optimization." In *Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence*, 62–67.
- [6] Dennis Herdiawan. 2021. "Indonesian Twitter Emotion." Kaggle.Com. 2021. <https://www.kaggle.com/datasets/dennisherdi/indonesian-twitter-emotion>.
- [7] Puteri, Dian Islamiaty. 2023. "Implementasi Long Short Term Memory (LSTM) Dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah." *Euler: Jurnal Ilmiah Matematika, Sains Dan Teknologi* 11 (1): (2002) The IEEE website. [Online], <http://www.ieee.org/>, tanggal akses: 16 September 2014.
- [8] Pradana, Yunico Ardian, Imam Cholissodin, and Diva Kurnianingtyas. 2023. "Analisis Sentimen Pemindahan Ibu Kota Indonesia Pada Media Sosial Twitter Menggunakan Metode LSTM Dan Word2Vec." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* 7 (5): 2389–97.
- [9] yuli Astari, Yuli, Afiyati Afiyati, and Saddam Wahib Rozaqi. 2021. "Analisis Sentimen Multi-Class Pada Sosial Media Menggunakan Metode Long Short-Term Memory (LSTM)." *Jurnal Linguistik Komputasional* 4 (1): 8–12.
- [10] Cahyani, Jumina, Syamsul Mujahidin, and Tegar Palyus Fiqar. 2023. "Implementasi Metode Long Short Term Memory (LSTM) Untuk Memprediksi Harga Bahan Pokok Nasional." *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)* 11 (2): 346–57.