ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 12, Nomor 3, Tahun 2023, Halaman 403 - 413
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

ANALISIS SENTIMEN VAKSIN COVID-19 PADA TWITTER MENGGUNAKAN *RECURRENT NEURAL NETWORK* (RNN) DENGAN ALGORITMA *LONG SHORT-TERM MEMORY* (LSTM)

Chintya Ayu Maharani[1*], Budi Warsito[2], Rukun Santoso[3]
[1,2,3] Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro
***e-mail**: chintyaaymhr@gmail.com

DOI: 10.14710/j.gauss.12.3.403-413

Article Info:
Received: 2022-12-14
Accepted: 2024-02-12
Available Online: 2024-02-26

Keywords:
Covid-19 Vaccine; Twitter; Sentiment Analysis; Recurrent Neural Network; Long Short-Term Memory

Abstract: The Coronavirus, also known as the Covid-19 pandemic, has reached every country worldwide, including Indonesia. Covid-19 is still prevalent and has killed many people in Indonesia. This makes it impossible to stop Covid-19 from spreading. The government's attempt to stop the Covid-19 pandemic is acquiring the vaccine. The administration of the Covid-19 vaccine has generated much discussion on social media, particularly Twitter. Tweets displaying public opinion on Twitter can be used for sentiment analysis and categorizing public opinion on the Covid-19 vaccine. 20,000 tweets were collected by Twitter crawling between January 10 and January 15, 2022. 3.290 tweets were left after pre-processing and meaningless tweets were eliminated. The data were processed using the *Recurrent Neural Network* method with the *Long Short-Term Memory* algorithm to determine its accuracy and identify topics often discussed by the public on Twitter. The LSTM method is capable of storing old information/data. A model with 70% training data, a learning rate of 0.01, 100 LSTM units, 32 batch sizes, 100 epochs, a cross-entropy loss function, and Adam optimizers was used to build the classification in this study. The accuracy value obtained from the performance evaluation of the *Long Short-Term Memory* model research was 80.34%.

1. PENDAHULUAN

Peristiwa pandemi Covid-19 menyebar ke seluruh belahan dunia, termasuk Indonesia. Covid-19 terus berada di kondisi yang tinggi ditandai dengan lonjakan kasus Covid-19 dan penurunan protokol kesehatan sehingga mengakibatkan banyak kematian di Indonesia. Salah satu varian Covid-19 yaitu *omicron* dinyatakan sebagai *Variant of Concern* (VOC) atau varian yang menjadi pusat perhatian oleh *World Health Organization* (WHO). Varian *omicron* merupakan varian baru yang tingkat penularannya paling cepat. Hal ini membuat penyebaran virus Covid-19 tidak dapat dikendalikan (Kemenkes, 2021). Pengadaan vaksinasi Covid-19 merupakan upaya pemerintah untuk mengatasi pandemi Covid-19. Terkait hal itu, terdapat beberapa opini pro dan kontra dari masyarakat terhadap efektivitas vaksin Covid-19 di Indonesia. Pengadaan vaksinasi Covid-19 ramai diperbincangkan di berbagai platform media sosial terutama Twitter. Opini masyarakat pada Twitter ditunjukkan dalam *tweet*, kumpulan *tweet* ini dapat digunakan dalam analisis sentimen dan pengklasifikasian opini masyarakat terkait vaksin Covid-19. Terdapat berbagai macam algoritma *deep learning* yang digunakan dalam melakukan klasifikasi teks, salah satunya adalah metode *Recurrent Neural Network* (RNN) dengan algoritma *Long Short-Term Memory* (LSTM) yang digunakan untuk memproses data berurutan sehingga dapat menyimpan informasi/data yang lama. Dalam analisis sentimen, metode LSTM telah banyak digunakan karena memberikan hasil yang lebih bagus daripada algoritma *machine learning*. Salah satu contoh penelitian analisis sentimen yang telah dilakukan adalah Murthy *et al.*

(2020) pada data IMDB dan Amazon *product* menggunakan algoritma *Long Short-Term Memory* (LSTM) mengenai ulasan film dan produk. Data yang digunakan sebanyak 50.000 ulasan, 25.000 di antaranya terpolarisasi positif dan 25.000 terpolarisasi negatif. Penelitan tersebut mendapatkan nilai akurasi pengujian sebesar 85%. Topik yang paling banyak dicari masyarakat Indonesia selama pandemi Covid-19 yaitu vaksin Covid-19. Oleh karena itu, pada penelitian ini dilakukan pembaruan topik yang dibahas yaitu vaksin Covid-19 dengan algoritma *Long Short-Term Memory*.

Berdasarkan latar belakang yang telah diuraikan, penelitian ini akan menggunakan *Recurrent Neural Network* dengan algoritma *Long Short-Term Memory* untuk analisis sentimen terkait vaksin Covid-19. Pengambilan data pada penelitian ini dilakukan dengan *Twitter Crawling*. Penelitian ini diharapkan dapat menentukan akurasi klasifikasi sentimen vaksin Covid-19 menggunakan *Recurrent Neural Network* dengan algoritma *Long Short-Term Memory* dan mengidentifikasi topik yang sering dibicarakan dalam sentimen positif dan sentimen negatif terkait vaksin Covid-19.

2. TINJAUAN PUSTAKA

Vaksin inaktif Covid-19 aman digunakan pada manusia karena menstimulasi sistem kekebalan tubuh tanpa risiko. Sistem kekebalan tubuh seseorang dapat berkembang secara alami terhadap suatu penyakit jika terpapar virus atau bakteri penyebab penyakit tersebut. Namun, infeksi virus Covid-19 memiliki risiko kematian dan daya tular yang tinggi. Sistem kekebalan tubuh perlu dibentuk vaksinasi. Vaksinasi Covid-19 bertujuan untuk melindungi masyarakat agar tetap produktif secara sosial dan ekonomi sekaligus menekan penyebaran penyakit, menurunkan angka kesakitan dan kematian akibat Covid-19, serta tercapainya kekebalan kelompok masyarakat (*herd immunity*) (Kemenkes, 2021).

Jejaring sosial Twitter mampu menyediakan sarana bagi pengguna untuk mengirim pesan yang panjangnya tidak lebih dari 280 karakter yang disebut *tweet*. Seiring berjalannya waktu, jejaring sosial ini memungkinkan pengguna untuk saling mengirim foto dan pesan teks. Proses pengambilan data dengan *Application Programming Interface* (API) Twitter baik berupa data pengguna maupun data *tweet* disebut *Twitter Crawling* (Sembodo *et al*., 2016). Pengguna harus memiliki *API key*, *API secret*, *Access Token*, dan *Access Token Secret* untuk keperluan akses dan ekstraksi *tweet*.

Proses mengolah data tekstual secara otomatis untuk mengidentifikasi dan mengkategorikan kalimat opini juga dikenal sebagai analisis sentimen atau *opinion mining*. Hasil dari analisis sentimen digunakan untuk mengidentifikasi opini positif dan atau negatif yang tersirat dalam teks (Liu, 2015). *Text mining* mengacu pada proses mengekstraksi informasi dari sumber data dengan mengidentifikasi dan mengeksplorasi pola yang menarik seperti klasifikasi teks, ekstraksi informasi, dan ekstraksi kata (Feldman dan Sanger, 2007). Solusi untuk masalah seperti memproses, menyortir, mengkategorikan, dan menganalisis data besar yang tidak terstruktur dapat ditemukan melalui *text mining* (Nurhuda dan Sihwi, 2014).

Transformasi data tekstual menjadi format yang lebih sederhana agar mudah dibaca oleh sistem sebagai persiapan untuk pemrosesan selanjutnya disebut sebagai *text pre-processing* (Indraloka dan Santosa, 2017). Tahap ini dilakukan agar pengolahan data dapat dibuat lebih efisien pada tahap awal *text mining*. Tahapan *text pre-processing* pada penelitian ini meliputi *case folding*, *remove* URL, *unescape* HTML, *remove mention*, *remove punctuation*, *remove number*, *remove duplicate*, dan normalisasi kata. *Tokenizing* melibatkan pemotongan dokumen menjadi potongan-potongan kecil menjadi satu yang disebut token dan terkadang diikuti langkah untuk membuang karakter tertentu seperti tanda

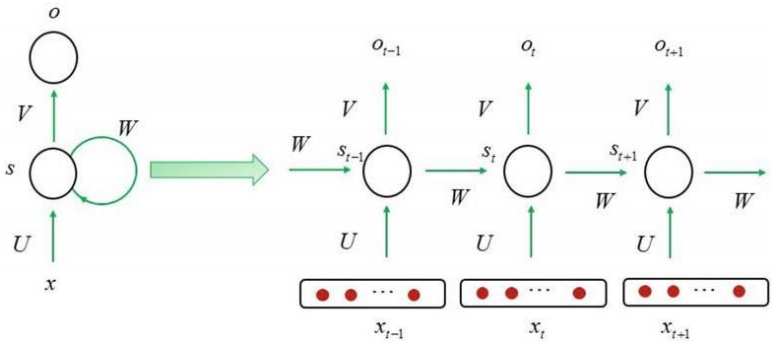
baca (Manning *et al*., 2009). Tanda baca dan karakter tidak penting lainnya dihilangkan dari dokumen teks sehingga analisis dapat berjalan lebih lancar.

Sentiment scoring bertujuan untuk melabelkan suatu pernyataan sehingga dapat digolongkan menjadi sentimen positif atau negatif berdasarkan kamus sentimen. Kamus yang digunakan adalah *Indonesian Sentiment Lexicon* (InSet) yang berisi kata-kata yang telah diberikan bobot sentimen 1 s.d. 5 untuk sentimen positif dan -1 s.d. -5 untuk sentimen negatif (Nielsen, 2011). InSet disusun menggunakan kumpulan kata dari *tweets* Indonesia yang terdiri dari 3.609 kata positif dan 6.609 kata negatif.

Stopwords removal berfungsi untuk menghilangkan kata – kata yang tidak diperlukan atau tidak memberikan kontribusi terhadap kesan atau pesan dalam suatu teks atau kalimat. Penggunaan *stopwords removal* terbukti dapat meningkatkan akurasi hasil sistem klasifikasi sentimen bila dibandingkan dengan tidak menggunakan *stopword* (Ghag dan Shah, 2015). Kata-kata berimbuhan juga tidak diperlukan dalam analisis sentimen. *Stemming* digunakan untuk mengubah kata berafiks, imbuhan, dan sufiks menjadi kata dasar dengan cara menghilangkannya. *Stemming* dilakukan untuk membuat varian kata yang memiliki arti yang sama menjadi satu varian kata yang seragam.

Teknik yang dapat mengubah kata-kata individual teks menjadi sebuah nilai vektor berupa bilangan riil dikenal sebagai *word embedding* (Brownlee, 2020). *Word embedding* memetakan kata ke dalam ruang vektor yang diwakili oleh vektor dan setiap kata dalam dokumen ke dalam vektor tersebut. Hasil *word embedding* berupa *lookup table* berbentuk matriks dengan *dictionary size* dan *embedding size*. *Dictionary size* merupakan ukuran kosakata dalam teks dan *embedding size* merupakan ukuran ruang vektor tempat kata-kata disematkan.

Recurrent Neural Network (RNN) merupakan proses yang dilakukan secara berulang untuk memproses *input* (umumnya data sekuensial) sebagai bagian dari *deep learning*. RNN pada Gambar 1 menggunakan struktur perulangan untuk meniru manusia dalam mengambil keputusan dengan menyimpan dan mengambil data historis sehingga dapat digunakan jika diperlukan (Li dan Qian, 2016).



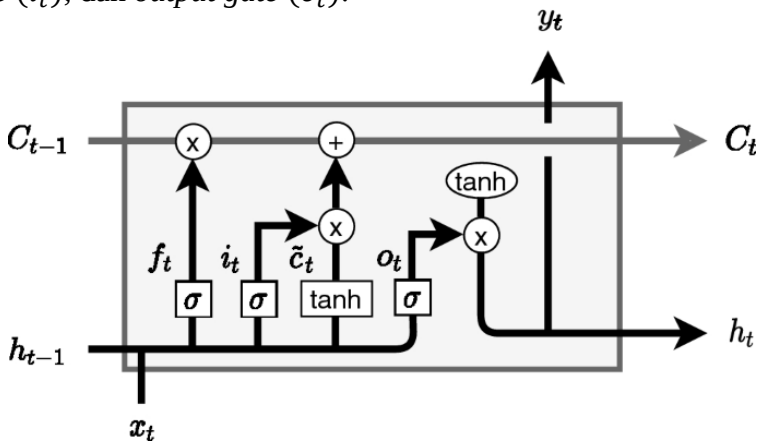
Sumber: Li dan Xu (2018)

Gambar 1. Arsitektur RNN

Simbol x_t merupakan *input* pada setiap langkah atau *time step* t, st merupakan *hidden state* pada setiap *time step* t, ot merupakan *output* untuk setiap *time step* t, dan U, V, W merupakan matriks parameter pada RNN.

Salah satu pengembangan *neural network* yang mampu mempelajari *long-term dependency* adalah *Long Short-Term Memory* (LSTM). Pada tahun 1997, Hochreiter dan Schmidhuber memperkenalkan LSTM yang semakin disempurnakan dan digunakan secara

luas hingga saat ini. LSTM merupakan variasi dari RNN yang dirancang untuk memperbaiki masalah memori jangka panjang pada RNN. LSTM memiliki tiga jenis *gates* yaitu *forget gate* (f_t), *input gate* (i_t), dan *output gate* (o_t).



Sumber: Torres *et al* (2022)

Gambar 2. Arsitektur LSTM

Langkah pertama pada LSTM adalah memilih informasi apa yang akan dibuang dari *cell state*,hal ini dibuat oleh *sigmoid layer* yang disebut *forget gate layer*. Pada gambar dapat dilihat sel LSTM akan memproses h_{t-1} dan x_t sebagai *input*. Langkah kedua dari *cell* LSTM adalah menentukan informasi yang akan disimpan di *cell state*. Proses ini memiliki dua bagian, pertama lapisan sigmoid menentukan nilai yang akan diperbarui dari *cell state*, lalu bagian kedua lapisan tanh membuat vektor dari kandidat baru, lalu keduanya digabung untuk melakukan pembaruan pada *cell state*. *Cell state* berfungsi untuk membawa informasi dari sel dibelakang ke sel-sel LSTM selanjutnya, pada setiap *timestep cell state* akan diperbarui dengan menggunakan *forget gate* dan *input gate* untuk menentukan informasi yang akan dibuang ataupun ditambahkan kedalam *cell state*. *Output gate* berguna untuk menentukan *output* dari *cell state* sekarang. Pertama, lapisan sigmoid menentukan bagian dari cell state yang menjadi output. Lalu, lapisan tanh akan mengubah nilai *cell state* menjadi antara -1 dan 1, kemudian nilai dari lapisan sigmoid dan lapisan tanh dikalikan (Olah, 2015). Berikut persamaan-persamaan yang digunakan pada sel LSTM:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (1)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1}) + b_c \quad (3)$$

$$C_t = i_t \circ \tilde{C}_t + f_t \circ C_{t-1} \quad (4)$$

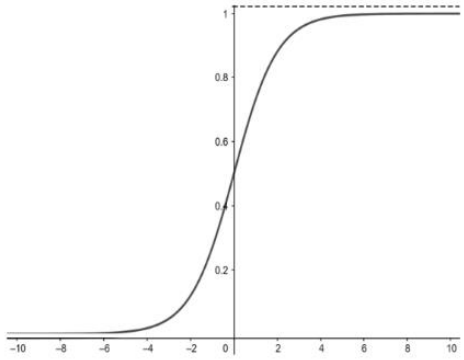
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \circ \tanh(C_t) \quad (6)$$

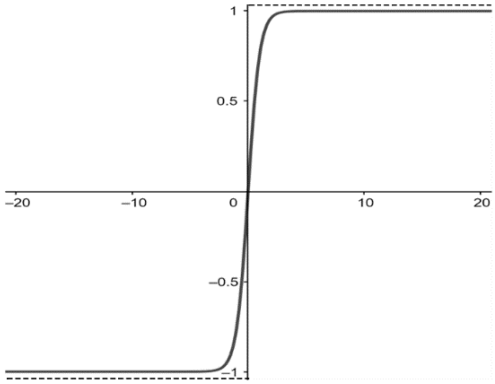
Simbol f_t merupakan forget gate, i_t merupakan input gate, $\tilde{C}_t$ merupakan *cell state candidate*. C_t merupakan *cell* state, o_t merupakan *output gate*, h_t merupakan *hidden state*, W merupakan matrik bobot, h_{t-1} merupakan *hidden state* sebelumnya, x_t merupakan data

input, b merupakan bias, σ merupakan fungsi aktivasi sigmoid, dan $tanh$ merupakan fungsi aktivasi tanh.

Sel LSTM menggunakan fungsi aktivasi sigmoid dan fungsi aktivasi tanh. Nilai input diubah ke dalam interval [0,1] oleh fungsi aktivasi sigmoid dan interval [-1,1] oleh fungsi aktivasi tanh. Gambar 2 dan 3 masing-masing menunjukkan grafik fungsi aktivasi sigmoid dan fungsi aktivasi tanh.



Gambar 3. Fungsi Aktivasi Sigmoid



Gambar 4. Fungsi Aktivasi Tanh

Persamaan 7 dan 8 adalah persamaan fungsi aktivasi sigmoid dan fungsi aktivasi tanh.

$$\sigma(x) = \frac{1}{1+e^{-x}}, -\infty < x < \infty \quad (7)$$

$$tanh(x) = \frac{2}{1+e^{-2x}} - 1, -\infty < x < \infty \quad (8)$$

Loss function merupakan metode untuk mengevaluasi seberapa baik algoritma memodelkan suatu data (Li *et al*., 2019). *Loss function* memiliki kurva yang bertujuan memberi tahu cara mengubah parameter untuk membuat model lebih akurat. Metode *cross entropy* termasuk *loss function* untuk masalah klasifikasi yang memiliki *output* berupa nilai probabilitas antara 0 dan 1.

$$L = -y \log(\hat{y}) - (1 - y)\log(1 - (\hat{y})) \quad (9)$$

Simbol y merupakan nilai target, dan $\hat{y}$ merupakan nilai prediksi.

Back Propagation menggunakan *error* untuk mengubah nilai bobot-bobotnya dalam arah mundur. *Error* didapatkan dengan menggunakan teknik perambatan maju (*Forward*

Propagation). *Back Propagation* memiliki tiga fase, yaitu fase maju (*Feed Forward*), fase mundur (*Back Propagation*), dan fase modifikasi bobot. Berikut ini merupakan persamaan untuk *back propagation*:

$$\delta h_t = \Delta_t + \Delta h_t \quad (10)$$

$$\delta o_t = \delta h_t * \tanh(C_t) * o_t * (1 - o_t) \quad (11)$$

$$\delta C_t = \delta h_t * o_t * \left(1 - tanh^2(C_t)\right) + \delta C_{t+1} * f_{t+1} \quad (12)$$

$$\delta \tilde{C}_t = \delta C_t * i_t * \left(1 - \tilde{C}_t^2\right) \quad (13)$$

$$\delta i_t = \delta C_t * \tilde{C}_t * i_t * (1 - i_t) \quad (14)$$

$$\delta f_t = \delta C_t * C_{t-1} * f_t * (1 - f_t) \quad (15)$$

$$\delta x_t = W^T * \delta gates_t \quad (16)$$

$$\Delta h_{t-1} = U^T . \delta gates_t \quad (17)$$

Perhitungan *back propagation* pada LSTM telah selesai, kemudian dilakukan perhitungan perubahan bobot dengan persamaan berikut :

$$\delta W = \sum_{t=1}^{T} \delta gates_t . x_t \quad (18)$$

$$\delta U = \sum_{t=1}^{T-1} \delta gates_{t+1} . h_t \quad (19)$$

$$\delta b = \sum_{t=1}^{T-1} \delta gates_{t+1} \quad (20)$$

Optimasi Adam digunakan untuk mengoptimalkan fungsi tujuan dalam *deep neural network*. Dalam metode ini, proses perubahan parameter tergantung pada gradien, *learning rate*, nilai momen pertama dan kedua dari gradien. Berikut ini merupakan persamaan untuk optimasi Adam (Zhang, 2018):

$$g_t = \nabla_\theta f(\theta_{t-1}) \quad (21)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (22)$$

$$v_t = \beta_2 m_{t-1} + (1 - \beta_2) g_t^2 \quad (23)$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (24)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (25)$$

$$\theta_{t+1} = \theta - \eta \frac{\widehat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (26)$$

Simbol θ_t merupakan parameter yang diperbaiki, η merupakan *learning rate*, $\widehat{m}_t$ merupakan bias dari estimator m_t, $\hat{v}_t$ merupakan bias dari estimator v_t, dan ϵ merupakan epsilon.

Evaluasi kinerja klasifikasi bertujuan untuk mengetahui seberapa baik klasifikasi yang telah dibuat. Tingkat akurasi prediksi klasifikasi yang dihasilkan dievaluasi sebagai bagian dari proses pengukuran. Evaluasi kinerja klasifikasi pada penelitian ini dilakukan dengan menggunakan *confusion matrix*. Keberhasilan klasifikasi dalam mengenali *tuple* dari kelas yang berbeda dapat dievaluasi menggunakan *confusion matrix* (Han *et al*., 2012). Parameter yang diperoleh dari *confusion matrix* untuk menilai kinerja klasifikasi yaitu *accuracy*, *precision*, dan *recall*. Indikator pada *confusion matrix* dapat dilihat pada tabel 1.

Tabel 1. *Confusion Matrix*

		Kelas Prediksi	
		Positif	Negatif
Kelas Aktual	Positif	TP	FN
	Negatif	FP	TN

Word Cloud merupakan metode untuk menampilkan representasi visual dari data teks dengan ukuran yang berbeda. Dalam *wordcloud*, semakin besar ukuran kata menunjukkan

semakin besar frekuensi kata muncul (Wardani *et al.*, 2019). Visualisasi menggunakan *word cloud* bertujuan untuk membantu pengamat dalam melihat gagasan atau kata yang sering muncul dengan tampilan yang menarik.

3. METODE PENELITIAN

Data yang digunakan pada penelitian ini merupakan data kualitatif yang diperoleh dari proses *crawling* Twitter pada tanggal 10-15 Januari 2022 dengan kata kunci "vaksin covid". *Tweet* yang didapatkan sebanyak 20.000 *tweet* dan berbahasa Indonesia. Variabel terikat pada penelitian ini adalah sentimen masyarakat mengenai vaksin Covid-19 dan variabel bebas adalah opini masyarakat terhadap vaksin Covid-19 pada Twitter yang ditunjukkan pada *tweet*. Data *tweet* yang diperoleh dibersihkan dari duplikat hingga tersisa 3.290 data.

Langkah-langkah analisis yang dilakukan pada penelitian ini adalah sebagai berikut:

1. *Twitter Crawling*
2. *Pre-processing* data
3. *Tokenizing*
4. *Sentiment Scoring*
5. *Stopwords removal*
6. *Stemming*
7. *Word embedding*
8. Klasifikasi dengan algoritma *Long-Short Term Memory*.
   a. Membagi data menjadi data latih (*training*) dan data uji (*testing*)
   b. Membangun model LSTM
   c. Mengevaluasi hasil kinerja klasifikasi menggunakan *confusion matrix*
9. Interpretasi dalam bentuk visual menggunakan *wordcloud*

4. HASIL DAN PEMBAHASAN

Tweet yang digunakan pada penelitian ini dikumpulkan dengan proses *twitter crawling* mulai dari tanggal 10 Januari 2022 hingga 15 Januari 2022 dengan *keyword* "vaksin covid" dengan kategori *tweet* berbahasa Indonesia sebanyak 20.000 *tweet*. Proses pengumpulan data dilanjutkan dengan menghilangkan duplikat dan *tweet* yang tidak memiliki arti hingga berkurang menjadi 3.290 *tweet*, data *tweet* yang akan diolah kemudian disimpan dalam format CSV (*Comma Separated Value*)

Pre-processing data dilakukan dengan tujuan mengubah data teks yang tidak terstruktur pada penelitian ini menjadi data yang terstruktur dan memudahkan tahap klasifikasi. Tahapan *pre-processing* yang dilakukan pada penelitian ini adalah sebagai berikut:

1. *Case Folding*
2. *Remove* URL
3. *Unescape* HTML
4. *Remove Mention*
5. *Remove Punctuation*
6. *Remove Number*
7. *Remove Duplicate*
8. Normalisasi kata

Normalisasi kata merupakan prosedur pengubahan kata tidak baku menjadi kata baku.

Pada proses *tokenizing*, *tweet* akan dipisah menjadi potongan kata dengan referensi pemisah berupa spasi. Hal ini dilakukan untuk memudahkan pelabelan data dalam *sentiment*

scoring. Proses pelabelan data pada penelitian ini dilakukan dengan *sentiment scoring* menggunakan *Indonesian Sentiment Lexicon* (InSet). Pelabelan data menggunakan kamus InSet *Lexicon* yang dilakukan program diperoleh 1.903 *tweet* berlabel negatif dan 1.387 *tweet* berlabel positif. Pelabelan data *tweet* secara manual menghasilkan 823 *tweet* (25%) berlabel negatif dan 2.467 *tweet* (75%) berlabel positif. Perbedaan pelabelan disebabkan oleh hasil skor akhir yang dihitung oleh program berbeda dengan perhitungan manual terutama untuk *tweet* yang berbeda makna. Data akhir yang digunakan adalah data *tweet* yang menggunakan pelabelan secara manual.

Stopwords removal bertujuan untuk menghapus kata-kata umum yang sering muncul dan tidak bermakna dengan menggunakan kamus *stopwords* Indonesia. *Stemming* dibutuhkan untuk meminimumkan indeks yang berbeda dengan membentuk kembali sebuah kata yang memiliki makna yang sama menjadi bentuk tunggal, seperti "ingatkan" menjadi "ingat", kemudian "penyebaran" menjadi "sebar". *Stopwords removal* dan *stemming* dilakukan untuk meningkatkan ketepatan hasil klasifikasi.

Lapisan pertama pada model LSTM yang akan dibangun adalah lapisan *word embedding*. Lapisan ini mengubah kata menjadi numerik seperti bobot nilai yang diinisialisasi secara acak menjadi *lookup table*. Lapisan *word embedding* terdapat beberapa parameter yaitu *input_dim* yang merupakan ukuran kosakata dalam teks yang memiliki ukuran 3.000 teks. Parameter lainnya yaitu *output_ dim* yang merupakan ukuran ruang vektor tempat kata-kata akan disematkan dengan ukuran 100 yang ditentukan dengan *trial error*. Parameter terakhir yaitu parameter *input_length* yang merupakan panjang dari urutan input yang berjumlah 17. Variabel *input* x_t berupa 3D dengan ukuran (3000, 17, 100) menjadi *output embedding* berupa 2D dengan ukuran (17, 100) yang selanjutnya akan dimasukan ke lapisan LSTM. Contoh hasil dari *word embedding* dapat dilihat pada Tabel 2.

Tabel 2. Contoh Hasil *Word Embedding*

Input Kata	Vektor
vaksin	[-0,02729131 0,02001158 -0,01437924 ⋯ -0,00627065 -0,00788884 0,0119754]
covid	[-0,00091212 -0,00304577 0,00002683 ⋯ 0,023942817 0,005896248 0,0300037]
booster	[-0,00091212 -0,00304577 0,00002683 ⋯ 0,023942817 0,005896248 0,0300037]

Data yang telah melalui tahapan *pre-processing* hingga *word embedding* dibagi menjadi data latih dan data uji. Perbandingan data latih dan data uji pada penelitian ini adalah sebesar 70%:30%. Pembangunan model LSTM dilakukan dengan cara *trial and error* pada setiap *hyperparameter* yang mungkin untuk mendapatkan model terbaik. *Epoch* ditentukan dengan nilai sebesar 100 *epoch* untuk semua model LSTM namun dapat berhenti sebelum mencapai 100 karena menggunakan *early stopping* untuk menghindari terjadinya *overfitting* yang terlalu parah. *Batch size* yang digunakan sebesar 32. Propagasi maju LSTM dihubungkan menggunakan jaringan *fully connected layer* sebanyak 1 unit dengan fungsi aktivasi sigmoid serta *optimizer* menggunakan Adam untuk *update* bobot dan *cross entropy* untuk mengetahui loss yang dihasilkan pada saat *training* data. Plan untuk *trial and error* tersebut dijabarkan pada Tabel 3.

Tabel 3. Plan Model LSTM

Trial	Units	*Learning Rate*
1	100	0,01
2	200	0,01
3	300	0,01
4	100	0,001
5	200	0,001
6	300	0,001
7	100	0,0001
8	200	0,0001
9	300	0,0001

Pemilihan model LSTM terbaik dilakukan setelah proses *training* dengan *trial and error* beberapa *hyperparameter* yang memungkinkan. Model LSTM terbaik ini nantinya digunakan sebagai model utama untuk melakukan prediksi pada data *testing*. Pemilihan model terbaik dari 9 Trial yang telah dilakukan sebelumnya, ditentukan berdasarkan model yang memiliki nilai *loss* validasi terendah. Perbandingan nilai *loss* pada masing-masing Trial dijabarkan pada Tabel 4.

Tabel 4. Perbandingan Nilai *Loss* pada 9 Trial

Trial	Units	*Learning Rate*	*Validation Loss*
1	100	0,01	**0,42897**
2	200	0,01	0,43311
3	300	0,01	0,47329
4	100	0,001	0,45423
5	200	0,001	0,44053
6	300	0,001	0,45243
7	100	0,0001	0,45737
8	200	0,0001	0,45046
9	300	0,0001	0,46898

Tabel 4 menunjukkan model terbaik yaitu model Trial 1 dengan *validation loss* terkecil sebesar 0,42897 dengan LSTM units sebesar 100 neuron dan *learning rate* sebesar 0,01. Selanjutnya model akan diuji kepercayaannya dengan dilakukan pengujian dengan data *testing*.

Model klasifikasi yang telah diuji dan dibangun menggunakan data latih akan dievaluasi kinerjanya. Evaluasi kinerja model pada penelitian ini menggunakan *confusion matrix*. Hasil dari *confusion matrix* untuk algoritma *Long Short Term-Memory* dapat dilihat pada tabel 5.

Tabel 5. *Confusion Matrix* algoritma *Long Short-Term Memory*

Kelas Aktual	Kelas Prediksi	
	Negatif	Positif
Negatif	123	120
Positif	74	670

$$Accuracy = \frac{123 + 670}{123 + 120 + 74 + 670} x100\% = 80{,}34\%$$

Hasil perhitungan *overall accuracy* menunjukkan tingkat akurasi model algoritma *Long Short Term-Memory* dalam mengklasifikasikan sentimen masyarakat terkait vaksin Covid-19 pada media sosial Twitter adalah sebesar 80,34%.

Visualisasi data teks berupa *tweet* dilakukan dengan menggunakan *word cloud*, visualisasi data dilakukan berdasarkan dua kelas sentimen yang telah ditentukan agar dapat diketahui kata-kata yang sering muncul pada masing-masing kelas sentimen. *Word cloud* untuk masing-masing kelas sentimen dapat dilihat pada Gambar 6.

(a) (b)

Gambar 5. (a) *Word Cloud* Kelas Sentimen Positif, (b) *Word Cloud* Kelas Sentimen Negatif

Topik atau kata dengan sentimen positif yang sering muncul dalam *tweet* yaitu kata "*booster*", "gratis" dan "dosis". Hal tersebut menunjukkan bahwa masyarakat pada media sosial Twitter banyak memberikan opini positif terkait dengan pemberian vaksin dosis ketiga (*booster*) secara gratis sedangkan kata dengan sentimen negatif yang sering muncul dalam *tweet* yaitu kata "*booster*", "tolak", dan "anak" kemudian dilanjutkan dengan kata "kena", "dosis", dan "mati". Hal tersebut menunjukkan bahwa masyarakat pada media sosial Twitter cenderung banyak memberikan opini negatif terkait dengan pasca vaksinasi dosis lengkap maupun belum lengkap (*booster*) yang masih bisa terkena infeksi virus Covid-19 dan adanya penolakan orang tua terhadap vaksinasi Covid-19 untuk anak karena kasus kematian pasca vaksin Covid-19.

5. KESIMPULAN

Analisis dan pembahasan mengenai analisis sentimen vaksin Covid-19 pada media sosial Twitter menunjukkan hasil bahwa klasifikasi menggunakan *Recurrent Neural Network* dengan algoritma *Long Short-Term Memory* menghasilkan akurasi sebesar 80,34%. Masyarakat Indonesia cenderung beropini positif terhadap vaksin Covid-19 dan hasil visualisasi *Wordcloud* menunjukkan bahwa topik yang sering dibahas dalam sentimen positif terkait vaksin Covid-19 yaitu mengenai pemberian vaksin *booster* Covid-19 secara gratis dari pemerintah sedangkan topik yang sering dibahas dalam sentimen negatif terkait vaksin Covid-19 yaitu mengenai kejadian pasca vaksin Covid-19 lengkap (*booster*).

DAFTAR PUSTAKA

Brownlee, J. 2020. *Long Short-Term Memory Networks With Python Develop Sequence Prediction Models With Deep Learning*. Machine Learning Mastery.

Feldman, R., dan Sanger, J. 2007. *The Text Mining Handbook: Advanced approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.

Ghag, K. V., dan Shah, K. 2015. *Comparative Analysis of Effect of Stopwords Removal on Sentiment Classification. International Conference on Computer, Communication and Control (IC4)*. India: Institute of Electrical and Electronics Engineers (IEEE).

Han, J., Kamber, M., dan Pei, J. 2012. *Data Mining: Concept and Techniques*. San Fransisco: Morgan Kaufmann Publishers.

Indraloka, D. S., dan Santosa, B. 2017. *Penerapan Text Mining untuk Melakukan Clustering Data Tweet Shopee Indonesia*. Jurnal Sains dan Seni ITS, 6(2): 6– 11. https://doi.org/10.12962/j23373520.v6i2.24419.

Kemenkes. 2021. *4 Manfaat Vaksin Covid-19 yang Wajib Diketahui*. https://upk.kemkes.go.id/new/4-manfaat-vaksin-covid-19-yang-wajib-diketahui.

Kemenkes. 2021. *Penjelasan WHO tentang Omicron, Varian Baru COVID-19*. https://covid19.go.id/p/berita/penjelasan-who-tentang-omicron-varian-baru-covid-19.

Li, C., Yuan, X., Lin, C., Guo, M., Wu, W., Yan, J., dan Ouyang, W. 2019. *AM-LFS: AutoML for loss function search*. Proceedings of the IEEE International Conference on Computer Vision, 2019-October(2), 8409–8418. https://doi.org/10.1109/ICCV.2019.00850.

Li, D., dan Qian, J., 2016. *Text Sentimen Analysis Based on Long Short-Term Memory*. Proceedings 1st IEEE International Conference on Computer Communication and the Internet. Wuhan, 13-15 Oktober, 471–475.

Li, S., dan Xu, J. 2018. A Recurrent Neural Network Language Model Based on Word Embedding. Springer, Cham, 368–377. https://doi.org/10.1007/978-3-030-01298-4_30.

Liu, B. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge: Cambridge University Press.

Manning, C., Raghavan, P., dan Schütze, H. 2009. *An Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Murthy, G. N., Allu, S. R., Andhavarapu, B., Bagadi, M. B. M. 2020. *Text based Sentiment Analysis using LSTM*. International Journal of Engineering and Technical Research V9(05). DOI: 10.17577/IJERTV9IS050290.

Nielsen, F. A. 2011. *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs*. CEUR Workshop Proceedings, 718(March 2011), 93–98.

Nurhuda, F., dan Sihwi, S. W. 2014. *Analisis Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naive Bayes Classifier*. ITSMART: Jurnal Ilmiah Teknologi Dan Informasi, 2: 35–42.

Olah, C., 2015. *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs.

Sembodo, J. E., Setiawan, E. B., dan Baizal, Z. A. 2016. *Data Crawling Otomatis pada Twitter*. Computational Science, School of Computing, Telkom University. October 2018, 11–16. https://doi.org/10.21108/indosc.2016.111.

Torres, J.F., Martínez-Álvarez, F. dan Troncoso, A. *A deep LSTM network for the Spanish electricity consumption forecasting*. Neural Comput dan Applic 34, 10533–10545 (2022). https://doi.org/10.1007/s00521-021-06773-2.

Wardani, F. K., Hananto, V. A., Nurcahyawati, V. 2019. *Analisis Sentimen Untuk Pemeringkatan Popularitas Situs Belanja Online di Indonesia Menggunakan Metode Naive Bayes (Studi Kasus Data Sekunder)*. JSIKA Vol. 08, No. 01.

Zhang, Z. 2018. *Improved Adam Optimizer for Deep Neural Networks*. IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 2018, pp. 1-2, doi: 10.1109/IWQoS.2018.8624183.