

Sentiment Analysis Komentar Berbahasa Bengkulu Menggunakan *Long Short-Term Memory (LSTM)*

Safitri Rahayu Nurhaeni^a, Ruvita Faurina^a, Ferzha Putra Utama^b, Kurnia Anggriani^b

^a Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu

^b Program Studi Sistem Informasi, Fakultas Teknik, Universitas Bengkulu

Jl. WR. Supratman Kandang Limun, Bengkulu 38371A, Indonesia

Informasi Naskah:

Diterima: 01 Juli 2023 / Direview: 02 Juli 2023/ Direvisi: 24 Juli 2023/ Disetujui Terbit: 25 Juli 2023

DOI: <https://doi.org/10.33369/pseudocode.10.2.117-125>

*Korespondensi: safitrinurhaenii@gmail.com

Abstract

Sentiment information is one type of information that can be obtained from social media. Sentiments can be interpreted as opinions and views of society which contain feelings. To analyze the value of sentiment whether the sentiment is a sentiment that tends to be neutral, negative, or positive, sentiment analysis can be used. language has its characteristics and uniqueness, Bengkulu language is no exception, because of this, it is necessary to model sentiment analysis for various languages. Sentiment modeling for the Bengkulu language is not yet available, therefore a sentiment analysis model for the Bengkulu language is developed by applying Long Short-Term Memory (LSTM), and architectural experiments for Long Short-Term Memory (LSTM) are carried out to obtain an architectural sentiment analysis model that produces the best value. The data used in the study amounted to 24,000 Bengkulu-language comments received from social media Instagram, Twitter, and Youtube. Experimental research 1 produces the best accuracy value compared to the results of testing in other experiments, with an accuracy value of 0.87 a precision value of 0.80, a recall value of 0.82, and an F1 score of 0.81.

Keywords: *Information, sentiment, Long Short-Term Memory (LSTM), Bengkulu Language, architecture, social media*

1. Pendahuluan

Tren untuk membuat berbagai jenis konten-konten di internet saat ini telah menjamur di kalangan masyarakat, berbagai jenis konten tersebut tersebar melalui sosial media, papan diskusi, forum *website* dan blog-blog. Bersamaan dengan tren tersebut, semakin banyak situs *website* di mana konsumen dapat menulis dan membaca ulasan, serta mengungkapkan pengalaman, peasant, pendapat, pandangan, dan keluhan mereka tentang berbagai produk dan layanan [1]. Sentimen dari pengguna menjadi hal penting dikarenakan jika ada yang ingin membeli produk baru, mendaftar kursus, atau ingin menonton film mereka akan melihat dan melakukan pengecekan terhadap *review*-nya terlebih dahulu dan berdasarkan *review* tersebut mereka membuat keputusan [2]. Salah satu aspek spesifik dari *platform* media sosial ialah *sentiment analysis* [3]. *Sentiment analysis* adalah tugas menganalisis pendapat orang dari sebuah teks untuk menentukan apakah sentimen itu positif, negatif, atau netral [4]. Selain itu *sentiment analysis* juga dapat diartikan sebagai proses menganalisa perasaan orang terhadap suatu aspek tertentu dengan menggunakan perhitungan matematis [5]. Mengidentifikasi sentimen dari pengguna adalah tugas yang sangat kompleks bahkan untuk manusia dan hal tersebut menjadi alasan mengapa dilakukan

pengembangan pada penelitian di bidang *sentiment analysis* [6].

Sudah banyak penelitian mengenai *sentiment analysis* dengan menggunakan berbagai jenis algoritma, arsitektur dan bahasa, seperti penelitian yang dilakukan oleh Lilis Kurniasari dan Setyanto (2020), Penelitian ini menggunakan metode *Long Short-Term Memory (LSTM)*, *Recurrent Neural Network (RNN)* untuk menganalisa sentimen pada *dataset* Bahasa Indonesia yang berjumlah 25.000 yang bersumber dari ulasan di Traveloka *website* dan Traveloka *mobile application*. Hasil dari percobaan menunjukkan bahwa *Recurrent Neural Network (RNN) - Long Short-Term Memory (LSTM)*, menghasilkan *accuracy* 95,0%, lebih baik dari CNN + *Word2vec*, Naïve Bayes, dan *Recurrent Neural Network (RNN) Konv* [7]. Penelitian lain yang dilakukan Ge Bin dkk (2018) mengkombinasikan *Chinese Sentiment Oriented dictionary* dan *Long Short-Term Memory (LSTM)*. Percobaan menunjukkan *accuracy* algoritma *Long Short-Term Memory (LSTM)* mencapai 93,51% [8]. Penelitian lainnya yang dilakukan oleh Basri Ciftci dan Mehmet Serkan Apaydin (2019) pada penelitian dilakukan perbandingan antar metode, yaitu metode Naïve Bayes, *Logistic Regression* dan metode *Long Short-Term Memory (LSTM)* pada *dataset review* berbahasa Turki. Penelitian, Naïve Bayes menghasilkan *accuracy* 80.2%, *Logistic Regression* menghasilkan *accuracy* 82.5% dan

Long Short-Term Memory (LSTM) menghasilkan *accuracy* 83.3% [9].

Berdasarkan penelitian-penelitian yang telah disebutkan sebelumnya, metode *Long Short-Term Memory* (LSTM) yang diterapkan pada *sentiment analysis* menghasilkan presentasi *accuracy* yang terbaik. *Long Short-Term Memory* (LSTM) merupakan jenis yang berbeda dari *Recurrent Neural Network* (RNN). *Long Short-Term Memory* (LSTM) dirancang untuk mengatasi masalah gradien yang menghilang atau *vanishing gradient* [10]. *Long Short-Term Memory* (LSTM) merupakan teknik *Deep Learning* yang dirancang untuk memproses *sentiment analysis*, pemodelan Bahasa, memprediksi data teks, serta menganalisa ucapan. *Long Short-Term Memory* (LSTM) dianggap sebagai bagian dari jaringan syaraf tiruan yang paling khusus dan memiliki kemampuan untuk mempelajari dan mengatasi long term dependencies yang diakibatkan oleh masalah *vanishing gradient* pada *Recurrent Neural Network* (RNN) dalam memproses suatu data text [11]. Dengan kata lain algoritma *Long Short-Term Memory* (LSTM) dikembangkan untuk membantu mengatasi komponen pertum buhan *gradient vector* yang berpotensi muncul pada algoritma *Recurrent Neural Network* (RNN) [7].

Bahasa Bengkulu sendiri adalah bahasa yang digunakan oleh masyarakat Bengkulu, terkhususnya masyarakat di Kota Bengkulu. Bahasa Melayu Bengkulu selain dipakai sebagai alat komunikasi dalam bersosialisasi sehari-hari digunakan dalam berbagai macam upacara adat seperti tabot, pernikahan, cukur rambut, kematian, ceramah keagamaan dan peringatan hari-hari besar tertentu [12]. Alasan khusus untuk keadaan ini, tidak ada penelitian yang menyelidiki Bahasa Bengkulu. *Sentiment analysis* pada Bahasa Bengkulu dilakukan untuk mengetahui apakah Bahasa Bengkulu yang digunakan merupakan Bahasa Bengkulu yang memiliki nilai berkonotasi negatif, netral, atau positif.

Penelitian ini akan dilakukan pengembangan model *sentiment analysis* dengan menerapkan *Long Short-Term Memory* (LSTM) serta dengan dilakukan percobaan pada arsitektur *Long Short-Term Memory* (LSTM) yang bertujuan untuk mencari *accuracy* paling baik yang kemudian akan diterapkan ke dalam model *sentiment analysis* komentar Bahasa Bengkulu.

2. Metodologi Penelitian

2.1 Materials

2.1.1 Sentiment Analysis

Sentiment analysis atau yang juga disebut *operation mining* adalah proses mengekstrak dan menganalisa opini orang, sentimen, sikap, persepsi, dan lain-lain. Terhadap beberapa entitas yang berbeda seperti layanan, topik dan produk. Evolusi aplikasi berbasis internet, situs *website*, jejaring media sosial, dan blog yang sangat cepat dapat mengarahkan orang untuk menghasilkan banyak opini dan ulasan terhadap produk, layanan, serta aktivitas sehari-hari [13]

Describe Ada banyak pernyataan dalam *sentiment analysis*. Pertama adalah, sudut pandang yang dianggap sebagai positif dalam satu kasus dapat dianggap negatif dalam kasus lain [14]. Dalam *sentiment analysis* terdapat

tiga tingkat klasifikasi: klasifikasi tingkat dokumen, klasifikasi tingkat kalimat dan tingkat aspek atau tingkat fitur. Di klasifikasi tingkat dokumen, klasifikasi tujuan utamanya adalah untuk mengklasifikasikan suatu opini dalam seluruh dokumen sebagai positif dan negatif. Dalam hal ini dispekulasikan bahwa seluruh dokumen tersebut adalah sebagai satu kesatuan. Tujuan klasifikasi tingkat kalimat adalah untuk mengkategorikan emosi yang diungkapkan masing-masing kalimat. Pada klasifikasi tingkat kalimat, langkah dasarnya adalah mengenali kalimat objektif atau subjektif. Misalkan terdapat kalimat subjektif, akan diputuskan apakah kalimat tersebut menyatakan pendapat negatif atau pendapat positif [14].

2.1.2 Bahasa Bengkulu

Bahasa Melayu Bengkulu selain dipakai sebagai alat komunikasi dalam bersosialisasi sehari-hari juga digunakan dalam berbagai macam upacara adat seperti Tabot, pernikahan, cukur rambut, kematian, ceramah keagamaan dan peringatan hari-hari besar tertentu. Penutur asli bahasa Melayu Bengkulu pada umumnya tinggal di pinggiran kota dan pesisir-pesisir pantai dikarenakan mata pencaharian mereka sebagai petani dan nelayan. Berdasarkan informasi yang penulis dapatkan, penutur-penutur asli bahasa Melayu Bengkulu yang terdapat di Kota Bengkulu antara lain terdapat di Pasar Bengkulu, Sukamerindu, Kampung Cina, sebagian Tanah Patah, sebagian di Sekip Kelurahan Kebun Kenanga, Berkas, Kebun Ros, Kuala Lempuing [12].

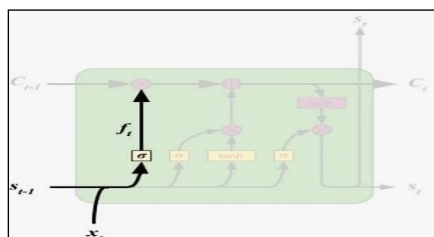
2.1.3 Long Short-Term Memory

Menurut Hochreiter dan Schmidhuber dalam Van Houdt dan rekan (2019) *Long Short-Term Memory* (LSTM) merupakan salah satu metode *Recurrent Neural Network* (RNN) *powerfull* yang didesain untuk mengatasi masalah ledakan/hilangnya *gradient* yang biasanya terjadi dalam proses *learning* jangka panjang, bahkan dapat terjadi pada jeda waktu minimal yang panjang [15]

Long Short-Term Memory (LSTM) terdiri dari tiga *gate* atau gerbang yang disebut *forget gate*, *input gate*, dan *output gate*.

2.1.3.1 Forget Gate

Forget gate menentukan bagian bagian dari C_{t-1} , di mana *cell state* diteruskan dari langkah sebelumnya seperti pada Gambar 1.



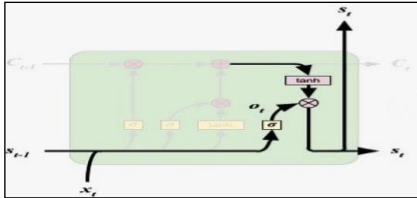
Gambar 1. Forget Gate Ft Dikalikan Dengan Cell State C_{t-1} Untuk Menyeleksi Forget Informasi

Pada Gambar 1 terjadi proses di mana *forget gate* F_t dikalikan dengan *cell state* C_{t-1} sudut pandang yang bertujuan untuk menyeleksi *forget information*. Cara ini seperti yang terdapat pada rumus (1) dapat dilakukan dengan melalui fungsi *gatekeeper sigmoid*.

$$f_t = \sigma(W_f [s_{t-1}, x_t] + b_f) \quad (1)$$

2.1.3.2 Input gate

Menurut Hochreiter dan Schmidhuber dalam Dautel dkk [16], *input gate* i_t , yang terdapat di Gambar 2.



Gambar 2. *Input gate* Mengarahkan *Cell state* Akan Di-update Kemana Dengan Nilai Kandidat Baru \tilde{C}_t

Gambar 2 merupakan proses di mana *input gate* akan mengarahkan *cell state* akan di-update dengan nilai kandidat yang baru. *Input gate* menggunakan sigmoid untuk mengontrol lajunya atau aliran informasi seperti pada rumus (2).

$$i_t = \sigma(W_i [s_{t-1}, x_t] + b_i) \quad (2)$$

Tujuan dari *input gate* adalah untuk melindungi informasi *Cell state*, yang terakumulasi selama langkah sebelumnya dari pembaruan yang tidak relevan. Oleh karena itu *gate* secara selektif memperbarui *cell state* dengan informasi yang [16]. Kemudian terdapat satu sel kandidat yang baru, \tilde{C}_t yang dihasilkan dengan mengaktifkan fungsi sepertipada rumus (3).

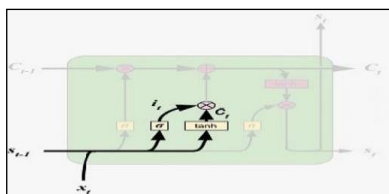
$$\tilde{C}_t = \tanh(W_c \cdot [s_{t-1}, x_t] + b_c) \quad (3)$$

2.1.3.3 Output Gate

Output gates mengarahkan prediksi *Long Short-Term Memory* (LSTM) sebenarnya, yang mana ditentukan oleh keduanya, yaitu *input* x_t dan *cell state* C_t seperti pada rumus (4).

$$o_t = \sigma(W_o [s_t, x_t] + b_o) \quad (4)$$

Output gate O_t terdiri atas sigmoid dengan argumen s_{t-1} dan x_t , dan menentukan informasi mana yang akan dilanjutkan ke *Layer* keluaran serta langkah *subsequent time* di *hidden state* s_t yang baru. Seperti yang digambarkan di Gambar 3 [16].



Gambar 3. *Output Gate* Yang Mengontrol Prediksi Jaringan

Gambar 3 merupakan proses di mana *output gate* akan mengontrol prediksi jaringan atau prediksi *Long Short-Term Memory* (LSTM) akan mengarah kemana [16].

2.1.4 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan metode yang dikenal baik dalam mengevaluasi pentingnya suatu kata dalam suatu dokumen [17]. Skor *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah kombinasi dari *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) sebagai $w_{ij} = tf_{ij} idf_j$ yang mana i adalah sebuah kata dalam kalimat j [9].

2.1.3 Word2vec

Semantic similarity pada kata memiliki peranan penting dalam bidang linguistik, terkhususnya yang memiliki keterkaitan dengan kesamaan arti kata-kata. Dalam hal ini, dua kata mungkin memiliki perbedaan sintaksis namun tetap memiliki arti yang sama [18]. *Word2vec* menghasilkan ruang vektor yang diperoleh dari korpus, yang terdiri dari kata-kata yang mirip dalam korpus dan adalah berdekatan satu sama lain di ruang *Word2vec* [19]. Terdapat dua algoritma *Word2vec* yaitu *Skip-Gram* dan *Continuous Bag-of-Word* (CBOW).

2.1.3.1 Continuous Bag-of-Word (CBOW)

Model ini memprediksi kata target dari kata-kata di sekitarnya *Continuous Bag-of-Word* (CBOW) memiliki waktu *training* lebih cepat dan memiliki *accuracy* yang lebih sedikit dibandingkan dengan *Skip-Gram* [20].

2.1.3.2 Skip-Gram

Model ini memprediksi kata-kata yang ada disekitarnya menggunakan kata target. *Skip-Gram* mempunyai *accuracy* representasi kata yang tinggi namun membutuhkan waktu yang lama untuk proses *training* [20].

2.2 Metode

2.2.1 Dataset Labelling

Penelitian ini menggunakan *dataset* berisi komentar berbahasa Bengkulu yang diperoleh dari 3 sosial media, yaitu: Instagram, Youtube, dan Twitter. Keseluruhan *dataset* berjumlah 24893 kalimat. Pelabelan *dataset* dilakukan secara manual dengan menggunakan 5 orang sebagai pelabel dan menggunakan *dataset* Bahasa Indonesia sebagai acuan pelabelan. Pelabelan ini menghasilkan *dataset* netral sebanyak 16821, *dataset* negatif 6.913, *dataset* positif 1.159. Meskipun ada beberapa keterbatasan yang ditentukan dalam perbandingan ini, namun dapat dijadikan sebagai indikator untuk memberikan pandangan umum tentang status proyek ini saat ini [21].

Tabel 1. Distribusi *Dataset*

No	Sentimen	Label	Total
1	Positif	1	8655
2	Negatif	-1	8299
3	Netral	0	8399

2.2.2 Pre-processing

Artikel Untuk mengurangi kesalahan komputasi, pada tahap *data cleaning* perlu dilakukan pengurangan jumlah fitur [22]. Data diproses terlebih dahulu untuk disiapkan untuk dianalisis dan kemudian melakukan prediksi sentimen [23]. Tahapan yang dilakukan diantaranya adalah sebagai berikut:

2.2.2.1 Remove Punctuation

Remove punctuation dilakukan untuk membersihkan data mentah yang memiliki beberapa karakter khusus, termasuk tagar, spasi putih berurutan, URL, dan simbol yang tidak perlu. Selain itu, ada sekumpulan emotikon yang kami bersihkan menggunakan sekumpulan representasi ikon yang telah ditentukan sebelumnya [24].

2.2.2.2 Tokenizing

Tokenizing dianggap sebagai langkah *pre-processing* yang penting dalam mengklasifikasikan data teks. Melatih algoritma untuk mengklasifikasikan data teks dengan menggunakan seluruh dokumen atau kalimat sangatlah sulit. Jadi, perlu dilakukan tokenisasi kalimat menjadi kata-kata dan melatih pengklasifikasi dengan kata positif, netral, dan negatif [25].

2.2.2.3 Case folding

Case folding adalah cara sederhana untuk membuat variasi kata menjadi lebih sedikit. Misalnya, jika ada dua kata "Uang" dan "uang" itu akan dikenali sebagai kata yang sama [26].

2.3 Word2vec

Kumpulan data yang diproses sebelumnya digunakan sebagai masukan ke *Word2vec* [26]. Model arsitektur *Word2vec* yang digunakan pada penelitian ini adalah model arsitektur *Continuous Bag-of-Word* (CBOW) dan *Skip-gram*. *Continuous Bag-of-Word* (CBOW) Arsitektur CBOW memprediksi kata saat ini berdasarkan konteksnya, dan *Skip-gram* memprediksi kata di sekitarnya dengan kata saat ini. Saat model *Continuous Bag-of-Word* (CBOW) dan *Skip-gram* dibandingkan terdapat perbedaan yang paling signifikan yaitu model yang menggunakan *Continuous Bag-of-Word* (CBOW) dapat memprediksi kemungkinan sebuah kata berdasarkan konteksnya. Konteksnya dapat memiliki sejumlah kata di dalamnya konteks *window* dapat digunakan untuk mengetahui seberapa besar kata disekitar kata yang digunakan sebagai konteks dan juga dapat menentukan jumlah kata yang terdapat di dalam suatu konteks [6]. Pada penelitian ini jumlah *window* yang digunakan

berjumlah 10 dan 5, *window* pada *Word2vec* merupakan jarak antara kata yang digunakan dengan prediksi dalam kalimat, semakin kecil nilainya maka semakin baik hasilnya. Model arsitektur ini digunakan dengan mempertimbangkan jumlah *dataset* yang sedikit dan ketidakseimbangan jumlah antara *dataset* negatif, positif, dan netral.

2.4 Pembentukan dan Pemilihan Model

Proses percobaan pada arsitektur pada *Word2vec* dan *Long Short-Term Memory* (LSTM) dilakukan untuk mengetahui arsitektur mana yang memiliki *accuracy* terbaik. Untuk menghasilkan model sentimen dilakukan proses *training* pada tiap-tiap model dengan *accuracy* yang berbeda-beda. Proses *training* dilakukan di Google Colabatory. Proses *training* ini memproses *dataset train* yang berjumlah sebanyak 80% dari keseluruhan *dataset*, *dataset train* ini digunakan sebagai masukan pada proses *training*. Proses *training* inilah yang kemudian akan menghasilkan model *sentiment analysis* Bahasa Bengkulu.

3. Hasil dan Pembahasan

Pada penelitian ini dilakukan percobaan terhadap arsitektur model *sentiment analysis* menggunakan *Long Short-Term Memory* (LSTM) dan *Word2vec*. Tahapan *training* dilakukan dengan menggunakan berbagai jenis arsitektur untuk mencari arsitektur yang sesuai dan memiliki nilai *accuracy* paling tinggi dan paling sesuai untuk model serta dataset sentiment dari komentar Berbahasa Bengkulu.

3.1 Arsitektur Word2vec

Table 1. Arsitektur Word2vec

Percobaan	Vector_size	Window	Workers	Min_count	sg	hs
1	100	10	4	1	0	1
2	150	10	4	1	0	1
3	100	10	4	1	1	1
4	100	10	4	1	0	0

Tabel 1. Arsitektur 1 *Word2vec* merupakan tabel yang berisikan 4 percobaan pada arsitektur 1 *Word2vec*, yang dimodifikasi adalah *vector_size*, penggunaan *skip-gram*, penggunaan *Continuous Bag-of-Word*, dan penggunaan *hierarchy softmax*.

3.2 vector_size

Pada penelitian ini nilai *vector_size* menggunakan nilai DIM (dimensi vektor) yang digunakan untuk menangkap nilai semantik dari korpus, besar dimensi yang digunakan bisa mempengaruhi *accuracy* sehingga dibutuhkan dimensi yang sesuai dan paling optimal.

3.2.1 Skip-Gram dan Continuous Bag-of-Word

SG merupakan *Skip-Gram*, jika nilai $SG = 0$, maka arsitektur menggunakan *Continuous Bag-of-Word (CBOW)*. Jika $SG = 1$ maka yang digunakan adalah arsitektur *Skip-Gram*.

3.2.2 Hierarchy Softmax.

Hierarchy Softmax berfungsi untuk meningkatkan efisiensi proses *training* pada dataset yang memiliki perpustakaan kosakata yang banyak. Bernilai 1 apabila menggunakan *Hierarchy Softmax* dan bernilai 0 apabila tidak menggunakan *Hierarchy Softmax*. Penggunaan ini dilakukan untuk mengetahui apakah benar efisiensi *training* dapat meningkat apabila digunakan *Hierarchy Softmax* pada dataset penelitian.

Table 2. Arsitektur 2 Word2vec

Percobaan	Vector_size	Window	Workers	Min_count	sg	hs
1	100	10	4	1	0	1
2	100	10	4	1	0	1
3	128	10	4	1	1	1
4	100	10	4	1	0	0

Tabel 2. Arsitektur 2 *Word2vec* merupakan tabel yang berisikan 4 percobaan pada arsitektur *Word2vec* untuk percobaan pada arsitektur 2 *Long Short-Term Memory (LSTM)*. Pada percobaan arsitektur 2 *Word2vec*, dilakukan modifikasi sama dengan modifikasi pada arsitektur 1 *Word2vec*, hanya saja dengan nilai yang berbeda.

3.3 Long Short-Term Memory

Table 3. Arsitektur 1 Long Short-Term Memory (LSTM)

N o	Em bed ding	D ro po ut	L S T M	De ns e 1	De ns e 2	De ns e 3	Ep oc h	Ba tc h siz e	In put len gth
1	100	0.	12	64	64	1	10	64	15
		2	8				0		0
2	150	0.	10	64	64	1	10	64	15
		2	0				0		0
3	100	0.	10	64	64	1	10	64	15
		2	0				0		0
4	100	0.	12	64	64	1	10	64	15
		2	8				0		0

Tabel 3. Arsitektur 1 *Long Short-Term Memory (LSTM)* merupakan tabel yang berisikan percobaan yang dilakukan pada arsitektur 1 *Long Short-Term Memory (LSTM)*. Pada arsitektur ini dilakukan 4 percobaan dengan mengubah beberapa nilai pada arsitektur *Long Short-Term Memory (LSTM)*, yang dimodifikasi dari arsitektur ini ialah nilai dari

output_dim di dalam *embedding layer* yang merupakan panjang vektor dalam setiap kata dan *Long Short-Term Memory (LSTM)* layer. Pada *embedding layer* yang *output_dim*-nya berjumlah sama dengan *vector_size* dari arsitektur 1 *Word2vec*.

Table 4. Arsitektur 2 Long Short-Term Memory (LSTM)

N o	Em bed ding	Drop out	L S T M 1	L S T M 2	De ns e	Ep oc h	Ba tc h Siz e	Inpu t Leng th
1	100	0.2	128	128	1	100	64	150
2	100	0.2	200	200	1	100	64	150
3	128	0.2	200	200	1	100	64	150
4	100	0.2	128	128	1	100	64	150

Tabel 4 Arsitektur 2 *Long Short-Term Memory (LSTM)* merupakan tabel yang berisikan percobaan yang dilakukan pada arsitektur 2 *Long Short-Term Memory (LSTM)*. Pada arsitektur ini dilakukan 4 percobaan dengan mengubah beberapa nilai pada arsitektur *Long Short-Term Memory (LSTM)*, yang dimodifikasi dari arsitektur ini ialah nilai dari *output_dim* di dalam *embedding layer* yang merupakan panjang vektor dalam setiap kata dan *Long Short-Term Memory (LSTM)* layer. Pada *embedding layer* yang *output_dim*-nya berjumlah sama dengan *vector_size* dari arsitektur 2 *Word2vec*.

3.4 Evaluasi

Hasil evaluasi didapatkan dengan dilakukannya pengujian Performa model sentimen dengan menggunakan *confusion matrix* yang berisikan empat nilai pengukuran performa dari masalah klasifikasi yang dilakukan pada model sentimen. Ada empat nilai atau poin yang ada pada *confusion matrix* yaitu *True Positif*, *True Negatif*, *False Positif* dan *False Negatif*. Setelah diketahui *confusion matrix*, maka akan diketahui nilai dari *accuracy* yang merupakan perhitungan untuk mengetahui seberapa tepat klasifikasi yang dibangun didalam model sentimen menyesuaikan dengan target yang telah ditetapkan, *precision* perhitungan untuk menguji *accuracy* data target dan nilai prediksi, *recall* merupakan perhitungan keberhasilan model, serta *F1-Score* menggambarkan perbandingan diantara *precision* dan *recall* dari model sentimen. Tabel 6 adalah hasil evaluasi model *sentiment analysis*.

Table 5. Hasil Pengujian Eksperimen 1

Percobaan	Metrics			
	Accuracy	Precision	Recall	F1-Score
1	0.84	0.84	0.84	0.84
2	0.85	0.85	0.85	0.85
3	0.84	0.84	0.84	0.84
4	0.86	0.86	0.86	0.86

Table 6. Hasil Pengujian Eksperimen 1

Percobaan	Metrics			
	Accuracy	Precision	Recall	F1-Score
1	0.86	0.86	0.86	0.86
2	0.84	0.84	0.84	0.84
3	0.85	0.85	0.85	0.85
4	0.84	0.84	0.84	0.84

Pada hasil *confussion matrix* di atas untuk percobaan pada arsitektur 1 di model percobaan keempat menghasilkan *accuracy* sebesar 0.86. Sedangkan untuk percobaan pada arsitektur 2, nilai *accuracy* tertinggi dihasilkan oleh percobaan pertama yang menghasilkan *accuracy* sebesar 0.86. Kedua arsitektur tersebut menghasilkan nilai *accuracy* yang sama, yaitu 0.86. Nilai *accuracy* ini digunakan untuk mengetahui seberapa tepat klasifikasi yang dibangun didalam model sentimen menyesuaikan dengan target yang telah ditetapkan. Pada penelitian ini data target yang digunakan merupakan *dataset* yang sudah diberikan label. Nilai *accuracy* kedua arsitektur menghasilkan nilai yang sama sehingga hasil dari proses *dataset sentiment* untuk model *sentiment analysis* tidak akan jauh berbeda.

6. Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dijabarkan *sebelumnya*, maka kesimpulan yang dapat diambil adalah penelitian ini telah menghasilkan sebuah *sentiment analysis* komentar Berbahasa Bengkulu dengan menggunakan *Long Short-Term Memory* (LSTM). Pengujian menggunakan *confussion matrix* menghasilkan nilai tingkat *accuracy* pada percobaan arsitektur 1 di model percobaan pertama sebesar 0,86 dan percobaan arsitektur 2 di model percobaan pertama sebesar 0.86..

Referensi

- M. A. Fauzi, "Random Forest Approach for *Sentiment analysis* in Indonesian Language Random Forest Approach fo *Sentiment analysis* in Indonesian Language," no. October 2018, pp. 46–50, 2019, doi: 10.11591/ijeecs.v12.i1.pp46-50.
- P. Shah, P. Swaminarayan, and M. Patel, "*Sentiment analysis* on film review in Gujarati language using machine learning," vol. 12, no. 1, pp. 1030–1039, 2022, doi: 10.11591/ijece.v12i1.pp1030-1039.
- A. Alrumaih, A. Al-sabbagh, R. Alsabah, H. Kharrufa, and J. Baldwin, "*Sentiment analysis* of comments in social media," vol. 10, no. 6, pp. 5917–5922, 2020, doi: 10.11591/ijece.v10i6.pp5917-5922.
- M. A. Fauzi, "Word2Vec model for *sentiment analysis* of product reviews in Indonesian language," vol. 9, no. 1, pp. 525–530, 2019, doi: 10.11591/ijece.v9i1.pp525-530.
- E. T. Khalid, E. B. Talal, M. K. Faraj, and A. A. Yassin, "*Sentiment analysis* system for COVID-19 vaccinations using data of Twitter," vol. 26, no. 2, pp. 1156–1164, 2022, doi: 10.11591/ijeecs.v26.i2.pp1156-1164.
- S. P. Panda, "*Sentiment analysis* using global vector and long short-term memory," vol. 26, no. 1, pp. 414–422, 2022, doi: 10.11591/ijeecs.v26.i1.pp414-422.
- L. Kurniasari and A. Setyanto, "*Sentiment analysis* using recurrent neural network-lstm in bahasa Indonesia," *J. Eng. Sci. Technol.*, vol. 15, no. 5, pp. 3242–3256, 2020, [Online]. Available: https://www.researchgate.net/publication/339846409_Sentiment_Analysis_using_Recurrent_Neural_Network.
- G. Bin, Z. Chong, H. Chunhui, and H. Yanli, "Classification algorithm of Chinese sentiment orientation based on dictionary and LSTM," *ACM Int. Conf. Proceeding Ser.*, pp. 119–126, 2018, doi: 10.1145/3291801.3291835.
- B. Ciftci and M. S. Apaydin, "A Deep Learning Approach to *Sentiment analysis* in Turkish," *2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018*, pp. 1–5, 2019, doi: 10.1109/IDAP.2018.8620751.
- H. K. Obayes, F. S. Al-turaihi, and K. H. Alhussayni, "Sentiment classification of user 's reviews on drugs based on global vectors for word representation and bidirectional long short-term memory recurrent neural network," vol. 23, no. 1, pp. 345–353, 2021, doi: 10.11591/ijeecs.v23.i1.pp345-353.
- U. Devi, G. Priyan, M. Kumar, and G. Chandra, "*Sentiment analysis* on Twitter Data by Using Convolutional," *Wirel. Pers. Commun.*, no. 0123456789, 2021, doi: 10.1007/s11277-021-08580-3.
- F. Senovil, "Morfofonemik Bahasa Melayu Bengkulu," *KLITIKA J. Ilm. Pendidik. Bhs. dan Sastra Indones.*, vol. 2, no. 2, pp. 165–178, 2020, [Online]. Available: www.journal.univetbantara.ac.id/index.php/klitika.
- M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on *sentiment analysis*: Approaches, challenges and trends," *Knowledge-Based Syst.*, vol. 226, p. 107134, 2021, doi: 10.1016/j.knosys.2021.107134.
- B. K. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative study of machine learning techniques in sentimental analysis," *Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2017*, no. Icicct, pp. 216–221, 2017, doi: 10.1109/ICICCT.2017.7975191.
- G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, 2020, doi: 10.1007/s10462-020-09838-1.
- A. J. Dautel, W. K. Hårdle, S. Lessmann, and H.-V. Seow, "Forex exchange rate forecasting using deep recurrent neural networks," *Digit. Financ.*, vol. 2, no. 1–2, pp. 69–96, 2020, doi: 10.1007/s42521-020-00019-x.
- R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the *sentiment analysis*," *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.
- D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in English words," *Procedia Comput. Sci.*, vol. 157, pp. 160–167, 2019, doi: 10.1016/j.procs.2019.08.153.
- P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "*Sentiment analysis* Using Word2vec and Long Short- Term Memory (LSTM) for Indonesian Hotel Reviews," *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.
- T. Onishi and H. Shiina, "Distributed Representation Computation Using CBOW Model and Skip-gram Model," *Proc. - 2020 9th Int. Congr. Adv. Appl. Informatics, IIAI-AAI 2020*, pp. 845–846, 2020, doi: 10.1109/IIAI-AAI50415.2020.00179.
- O. J. Ying, M. Mun, N. Ramli, and U. U. Sheikh, "*Sentiment analysis* of informal Malay tweets with deep learning," vol. 9, no. 2, pp. 212–220, 2020, doi: 10.11591/ijai.v9.i2.pp212-220.
- E. Sutoyo and A. Almaarif, "Twitter *sentiment analysis* of the relocation of Indonesia 's capital city," vol. 9, no. 4, pp. 1620–1630, 2020, doi: 10.11591/eei.v9i4.2352.
- H. Khalid and K. Hasan, "Sentiment classification of user 's reviews on drugs based on global vectors for word representation and bidirectional long short-term memory recurrent neural network Sentiment classification of user 's reviews on drugs based on global vectors for word representation and bidirectional long short-term memory recurrent neural network," no. July, 2021, doi: 10.11591/ijeecs.v23.i1.pp345-353.
- M. Maree, M. Eleyat, S. Rabayah, and M. Belkhatir, "A hybrid composite features based sentence level sentiment analyzer," vol. 12, no. 1, pp. 284–294, 2023, doi: 10.11591/ijai.v12.i1.pp284-294.
- V. D. Chaitra, "Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments," vol. 9, no. 5, pp. 4452–4459, 2019, doi: 10.11591/ijece.v9i5.pp4452-4459.
- F. Nurifan, R. Sarno, and C. S. Wahyuni, "Developing Corpora

using Word2vec and Wikipedia for Word Sense Disambiguation,” vol. 12, no. 3, pp. 1239–1246, 2018, doi: 10.11591/ijeecs.v12.i3.pp1239-1246.

29. A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, “Multinomial Naive Bayes for Text Categorization Revisited,” pp. 488–499, 2004.
30. L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, “Application of Deep Learning in Food: A Review,” *Compr. Rev. Food Sci. Food Saf.*, vol. 18, no. 6, pp. 1793–1811, 2019, doi: 10.1111/1541-4337.12492.
31. M. Hassan, S. Shakil, N. N. Moon, and M. M. Islam, “Sentiment analysis on Bangla conversation using machine learning approach,” vol. 12, no. 5, pp. 5562– 5572, 2022, doi: 10.11591/ijece.v12i5.pp5562-5572.
32. E. M. Hambli and F. Benabbou, “A deep learning based technique for plagiarism detection : a comparative study,” vol. 9, no. 1, pp. 81–90, 2020, doi: 10.11591/ijai.v9.i1.pp81-90.
33. C. Romli, Ikhsan;Kharida, Fairuz; Naya, “Penentuan Kepuasan Pelanggan Terhadap Pelayanan Kantor Pelayanan,” *J. RESTI*, vol. 1, no. 10, 2021.