

# IMPLEMENTASI LONG SHORT-TERM MEMORY DALAM ANALISIS SENTIMEN PENGGUNA APLIKASI TWITTER YANG MENGANDUNG UJARAN KEBENCIAN

Lucky Yosia Wibowo, Norul Annisa, Puteri Ananda Khairunnisa

Viktor Handrianus Pranatawijaya, Ressa Priskila

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya, Palangka Raya

puteriananda11@gmail.com

## ABSTRAK

Aplikasi dan media sosial muncul sebagai hasil dari kemajuan internet dan teknologi, yang memungkinkan pengguna berinteraksi satu sama lain melalui teks dan komentar. Sebagai contoh, aplikasi Twitter merupakan media sosial yang sering sekali digunakan saat ini. Namun, Twitter juga berpotensi menjadi tempat penyebaran ujaran kebencian. Ujaran kebencian berarti menghina, merendahkan, atau mengintimidasi seseorang atau kelompok berdasarkan atribut tertentu, seperti ras, agama, etnis, gender, orientasi seksual, atau jenis kelamin. Ujaran kebencian memiliki banyak akibat buruk yang dapat membahayakan kesehatan mental dan keamanan online penggunanya. Tujuan penelitian ini adalah untuk memecahkan masalah ini dengan menguji seberapa efektif algoritma *Long Short-term Memory* (LSTM) dalam menganalisis sentimen ujaran kebencian pada aplikasi. Untuk mendeteksi ujaran kebencian dengan tingkat akurasi tinggi, penelitian ini mengembangkan model klasifikasi teks berbasis LSTM. Data teks yang dikumpulkan dari aplikasi yang mengandung ujaran kebencian kemudian dibersihkan dan dinormalisasi untuk pra-pemrosesan. Selain itu, model LSTM digunakan untuk klasifikasi teks sentimen, dan kinerjanya dinilai dengan metrik *accuracy*, *precision*, dan *recall*. Sehingga penelitian ini menghasilkan tingkat *accuracy* sebesar 83% dengan nilai rata-rata makro (*macro average*) sebesar 65%.

**Kata kunci :** analisis sentimen, kecerdasan buatan, klasifikasi teks, long short-term memory, peringkasan teks

## 1. PENDAHULUAN

Di era komputer dan internet saat ini, ada banyak informasi, dan mengolahnya dengan cepat dan efisien menjadi sangat penting. Peringkasan teks otomatis mempersingkat waktu yang dibutuhkan untuk membaca teks yang sangat panjang karena dapat memberikan informasi dalam waktu yang sangat singkat. [1] Kemampuannya untuk menghasilkan ringkasan abstrak yang informatif dan koheren telah ditunjukkan oleh algoritma LSTM, sebuah jaringan saraf tiruan, yang memiliki kemampuan untuk menangani data sekuensial, seperti teks.

Tujuan dari penelitian ini adalah untuk mengembangkan model LSTM yang dapat diandalkan untuk menghasilkan ringkasan teks abstrak berkualitas tinggi. LSTM memiliki mekanisme yang lebih mengedepankan memori daripada langkah pelatihan sebelumnya untuk menghasilkan hubungan antar kata, yang membuatnya ideal untuk memproses text speech, translation, dan summarization. [2]

Penelitian ini berjudul "Analisis Sentimen Evaluasi Pembelajaran Menggunakan Metode Long Short-Term Memory" yang bertujuan untuk melakukan analisis sentimen menggunakan metode Long Short-Term Memory (LSTM) dalam konteks Bahasa Indonesia. Teknik abstraksi bertujuan untuk membuat kalimat yang menguraikan cara manusia mengambil intisari dari dokumen yang dibaca.[1] Jika dibandingkan dengan metode ekstraktif, metode abstrak dapat menghasilkan ringkasan yang dinilai lebih tepat, komplit, dan bermakna. Metode ini juga dapat mengurangi redundansi pada konsep dan hubungan antar kata, sehingga ringkasan dapat lebih tepat sesuai konteks. [3]

## 2. TINJAUAN PUSTAKA

Peringkasan teks adalah proses menghasilkan ringkasan dari sebuah teks yang panjang. Ringkasan yang baik dapat membantu pembaca memahami esensi teks tanpa harus membaca keseluruhan teks. Ada dua jenis ringkasan teks: ekstraktif dan abstraktif. Peringkasan ekstraktif memilih kalimat-kalimat penting dari teks asli, sedangkan peringkasan abstraktif menghasilkan ringkasan baru dengan menggunakan kata-kata yang berbeda dari teks asli.[4]

### 2.1. Long Short-term Memory

LSTM adalah jaringan saraf tiruan (JST) yang dimaksudkan untuk menangani data sekuensial, seperti teks. Dengan kemampuan untuk mempelajari ketergantungan jangka panjang dalam data sekuensial, LSTM dapat menghasilkan ringkasan teks yang lebih akurat dan informatif daripada metode tradisional. Struktur Long Short-term Memory (LSTM) dan Recurrent Neural Network (RNN) memiliki bentuk rangkaian modul berulang dari jaringan syaraf tiruan. Namun, struktur LSTM memiliki empat lapisan yang saling berinteraksi, seperti lapisan tunggal Tanh.[11]

### 2.2. Sentiment Analysis

Sentiment Analysis atau analisis sentimen adalah proses mempelajari data teks secara otomatis untuk menghasilkan informasi yang terkandung dalam kalimat. Analisis sentimen biasanya digunakan untuk melihat perspektif suatu kalimat terhadap sebuah masalah, apakah berpendapat positif atau negatif. Salah satu contoh penggunaan analisis sentimen adalah menemukan kecenderungan pasar dan pendapat

pasar tentang suatu barang.[12] Sentiment Analysis merupakan bagian dari ilmu Natural Language Processing (NLP) dan bergerak secara berkelanjutan dimulai dari tahap klasifikasi teks, hingga tahap mereview polaritas atau perbedaannya.[13]

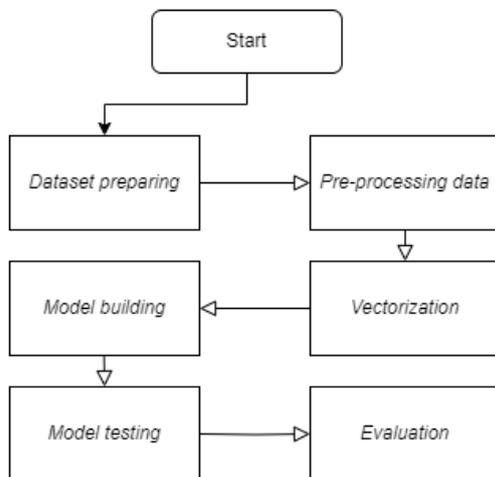
### 2.3. Text Summarizer

Salah satu cabang ilmu pemrosesan bahasa alamiah (Natural Language Processing) adalah peringkasan teks. Peringkasan dokumen teks otomatis, juga dikenal sebagai peringkasan teks otomatis, adalah metode untuk mendapatkan intisari informasi dari sebuah atau lebih dokumen teks.[14] Peringkasan teks dapat dilakukan dalam dua cara: peringkasan ekstraktif (ekstraktif peringkasan) dan peringkasan abstrak (abstraktif peringkasan). Peringkasan ekstraktif menghitung bobot setiap kata dan mengambil kata-kata dengan bobot tertinggi sebagai ringkasan. Peringkasan abstrak membuat dan menyusun kalimat baru sebagai hasil dari peringkasan yang dibuat sebelumnya. [4]

Berikut adalah persamaan yang digunakan dalam LSTM

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [s_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [s_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W_c \cdot [s_{t-1}, x_t] + b_c) \\
 C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
 o_t &= \sigma(W_o \cdot [s_{t-1}, x_t] + b_o) \\
 s_t &= o_t * \tanh(C_t)
 \end{aligned}
 \tag{1}$$

### 3. METODE PENELITIAN



Gambar 1. Flowchart LSTM

Metode ini dimulai dengan menyiapkan kumpulan data untuk digunakan sebagai data latih. Dataset yang digunakan adalah Sentiment Analysis Aplikasi Twitter yang mengandung ujaran kebencian. Dataset ini diambil dari salah satu artikel yang diunggah pada situs web Kaggle. Metode LSTM dipilih karena keberhasilannya dalam peringkasan abstraktif dalam bahasa Inggris. Pada tahap terakhir,

model yang telah dibuat akan dievaluasi untuk keakuratannya.

### 3.1. Data preparing

Data yang digunakan adalah *dataset* dari salah satu situs web penyedia dataset, Kaggle dalam artikel yang ditulis oleh Analytics Vidhya. Tujuan dari tugas ini adalah untuk mendeteksi ujaran kebencian dalam *tweet*, bahwa sebuah *tweet* mengandung ujaran kebencian jika *tweet* tersebut memiliki sentimen rasis atau seksis yang terkait dengannya. Jadi, tugasnya adalah mengklasifikasikan *tweet* rasis atau seksis dari *tweet-tweet* lainnya.

Secara formal, dengan diberikan sebuah sampel pelatihan *tweet* dan label, di mana label '1' menunjukkan *tweet* yang mengandung ujaran kebencian dan label '0' menunjukkan *tweet* tersebut tidak mengandung ujaran kebencian, tujuannya adalah untuk memprediksi label pada dataset uji.

Tabel 1. Dataset

Dataset	Index
"@user hey, white people: you can call people 'white' by @user #race #identity #med🇲🇵🇲🇵🇲🇵"	1
i am thankful for saturdays. #thankful #positive	0
#fathersday to all our the amazing fathers out there today.	0
#stereotyping #prejudice offer no #hope or solutions but create the same old repetitive #hate #conflict🇲🇵🇲🇵🇲🇵	1
packing sneak-peek! to be working with the #yellowjackets of the #neshl for 2016! @user #bhl #hockey	0

Kata-kata yang mengacu pada hal rasis seperti warna kulit atau suatu golongan tertentu, contohnya : "black", "white", dan "racist". Juga kata-kata yang mengarah pada ujaran kebencian seperti : "hate", "hatred", dan "retard".

### 3.2. Pre-processing data

Menyiapkan jumlah data yang akan digunakan oleh komputer sebagai bahan pembelajaran adalah langkah pertama dalam pembuatan program. Data ini berfungsi sebagai referensi untuk model yang akan dibuat. Tahap yang akan dilakukan pada pra-proses adalah tokenisasi, mengubah kata menjadi lower case, stopword removal.[5]

#### a. Casefolding

Casefolding adalah tahapan untuk mengubah huruf besar menjadi huruf kecil. Contohnya, "mAkANaN" menjadi "makanan".

#### b. Tokenization

Tokenization atau tokenisasi adalah tahapan pemecahan kalimat menjadi himpunan kata-kata. Contohnya [" pagi hari ini aku sudah makan buah apel."] Menjadi ["pagi", "hari", "ini", "aku", "sudah", "makan", "buah", "apel"].

c. *Stopword removal*

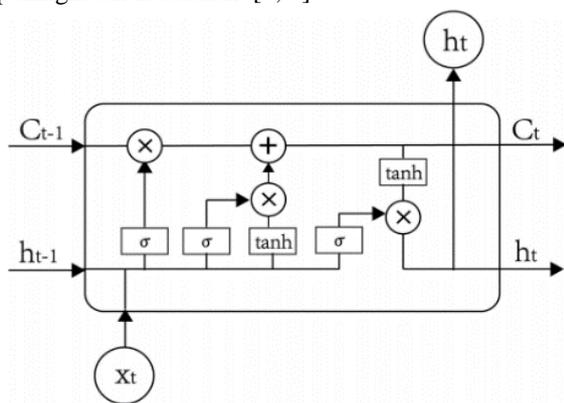
Stopword removal adalah tahapan pengeliminasian kata-kata yang tidak penting. Contohnya, [“pagi”, “hari”, “ini”, “aku”, “sudah”, “makan”, “buah”, “apel”] menjadi [“pagi”, “ini”, “aku”, “makan”, “buah”, “apel”].

3.3. *Vectorization*

Pada langkah ini, vektor dibuat dari kata-kata yang telah dibersihkan dan siap digunakan. Ini dilakukan agar kata-kata dapat dihitung dan memiliki nilai persamaan antara satu kata dan yang lain. Vektor yang telah diolah sebelumnya digunakan untuk penelitian ini. Dalam penelitian ini menggunakan GloVe sebagai word embedding vector.

3.4. *Model building*

Metode yang digunakan adalah dengan menggunakan Jaringan Syaraf Tiruan, juga dikenal sebagai Neural Network. Sistem kerja Jaringan Syaraf Tiruan mirip dengan sistem kerja otak manusia. Setiap jaringan memiliki layer input, layer hidden, dan layer output. [6] Model dibangun menggunakan metode bi-directional RNN atau perhitungan bobot RNN secara bolak-balik. LSTM adalah variansi RNN yang digunakan, yang memiliki kemampuan untuk mengolah data lebih akurat. [7] Salah satu kelebihan LSTM daripada RNN konvensional adalah arsitektur yang mengingat dan melupakan output yang akan diproses kembali menjadi input. Kemampuan tambahan LSTM adalah mempertahankan kesalahan yang terjadi selama backpropagation, yang mencegah peningkatan kesalahan. [7, 8]



Gambar 2. Model LSTM

Model LSTM ini sudah teruji dapat menyelesaikan beberapa kasus NLP, seperti machine translation [9] dan text summarization [10].

4. **HASIL DAN PEMBAHASAN**

Penelitian ini melihat bagaimana algoritma Long Short-term Memory (LSTM) dapat digunakan untuk menganalisis sentimen ujaran kebencian pada aplikasi Twitter. Tujuannya adalah untuk membuat model klasifikasi teks yang baik untuk menemukan ujaran kebencian. Data teks dikumpulkan dari aplikasi yang berisi ujaran kebencian. Setelah itu, data dibersihkan

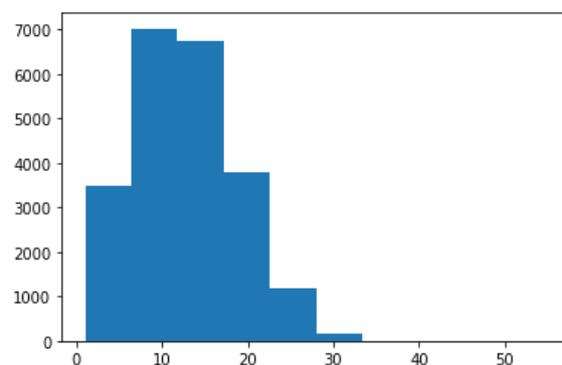
dan dinormalisasi untuk digunakan sebelum diproses. Untuk klasifikasi teks sentimen, model LSTM juga akan digunakan, dan untuk meningkatkan kualitas kinerja, parameter model perlu untuk dioptimalkan. Model LSTM menunjukkan hasil yang lebih baik dibandingkan model klasifikasi teks lain dalam mendeteksi ujaran kebencian. Model ini mampu mengidentifikasi berbagai jenis ujaran kebencian dengan tingkat akurasi tinggi.

Tabel 2. Tabel sentimen prediksi

Sentimen Prediksi	
Sentimen Positif	Sentimen Negatif
31962	2242

Setelah selesai dilakukan testing menggunakan data testing, dari 31962 evaluasi maka didapat bahwa terdapat 29720 evaluasi komentar yang diprediksi bersentimen positif dan kemudian terdapat 2242 evaluasi komentar yang diprediksi bersentimen negatif. Pertama melakukan analisis data pada panjang urutan tweet untuk mempersiapkan data untuk model jaringan saraf. Panjang urutan bervariasi, sehingga tidak memungkinkan untuk menggunakan array yang sama. Kemudian membuat daftar panjang urutan dan menganalisis distribusinya menggunakan fungsi histogram dan panda, dan mengisi urutan dengan angka nol agar semuanya memiliki panjang yang sama.

Lalu menggunakan fungsi NumPy deep copy untuk membuat salinan data, kemudian menggunakan fungsi enumerate Python untuk mengulang data dan mengisi setiap urutan dengan angka nol hingga mencapai panjang yang diinginkan. Terakhir menyiapkan data validasi dan pengujian dengan cara yang sama. Setelah langkah prapemrosesan data ini, pembicara beralih ke pendefinisian dan pelatihan model jaringan syaraf tiruan.



Gambar 3. Histogram sequence\_length

Bisa dilihat mayoritas nilainya berada pada angka 0 hingga 35, dan mungkin ada beberapa nilai yang berada pada angka 40-55. Dan juga angka maksimum berada pada angka 55, yang berarti tidak ada satupun nilai pada angka yang melebihi 55 token.

Tabel 3. Data sequence\_length

Data Sequence	
count	22373.000000
mean	12.692308
std	5.929912
min	1.000000
25%	8.000000
50%	12.000000
75%	17.000000
max	55.000000

Kemudian mendefinisikan model dari sequential dan menambahkan arbitrary number, menggunakan teknik pemerataan dengan Flatten, dan menambahkan nilai sigmoid menjadi 1 agar mendekteksi inputan yang mengandung ujaran kebencian.

Tabel 4. Hasil model sequential

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 57, 64)	29440
dropout (Dropout)	(None, 57, 64)	0
lstm_1 (LSTM)	(None, 57, 64)	33024
dropout_1 (Dropout)	(None, 57, 64)	0
lstm_2 (LSTM)	(None, 57, 64)	33024
dropout_2 (Dropout)	(None, 57, 64)	0
flatten (Flatten)	(None, 3648)	0
dense (Dense)	(None, 1)	3649
Total params: 99,137 Trainable params: 99,137 Non-trainable params: 0		

Setelah selesai dilakukan training data frequency pada model Sequential, didapat bahwa terdapat 20820 evaluasi komentar secara aktual bersentimen positif dan kemudian terdapat 1553 evaluasi komentar secara aktual bersentimen negatif.

Tabel 5. Sentimen aktual

Sentimen Aktual	
Sentimen Positif	Sentimen Negatif
20820	1553

Dari data tabel diatas, maka dapat dicari nilai *precision*, *recall*, *accuracy*, dan *f-measure* sebagai berikut :

Tabel 6. Hasil test prediction

	precision	recall	f1-score	support
0	98%	83%	90%	4454
1	26%	80%	39%	341
accuracy			83%	4795
macro avg	62%	81%	65%	4795
weighted avg	93%	83%	86%	4795

*Accuracy* merupakan salah satu metrik penting untuk mengukur kinerja model klasifikasi. Sederhananya, *accuracy* menunjukkan seberapa sering model membuat prediksi yang benar. *Accuracy* berisi rasio prediksi benar (positif dan negatif) dengan

keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

*Precision* adalah ukuran tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model, khususnya untuk tugas klasifikasi. *Precision* mencakup rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif, menghitung jumlah data dari semua kelas yang telah diprediksi dengan benar, berapa banyak data yang benar-benar positif.

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

*Recall* adalah ukuran keberhasilan model dalam menemukan kembali informasi. Ini mencakup rasio prediksi benar positif dibandingkan dengan total data benar positif.

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

## 5. KESIMPULAN DAN SARAN

Hasil percobaan pada model testing LSTM untuk analisis sentimen pada cuitan pengguna aplikasi Twitter menunjukkan bahwa model dengan batch size 64 dan nilai epoch 20 mendapatkan nilai *accuracy* sebesar 83% dan nilai *f-measure* sebesar 65%. Nilai tersebut berarti sistem sudah cukup dapat menghasilkan prediksi sentimen yang sesuai dengan data latih. Untuk bagian tweet yang tidak mengandung ujaran kebencian memiliki nilai *precision* sebesar 98%, nilai *recall* sebesar 83%, dan nilai *f1-score* sebesar 90%. Untuk bagian tweet yang mengandung ujaran kebencian memiliki nilai *precision* sebesar 26%, nilai *recall* sebesar 80%, dan nilai *f1-score* sebesar 39%. Untuk model ini, kita mendapatkan hasil rata-rata makro dari *f1*-nya yang berjumlah 65% saja, namun sebagai saran jika beberapa bagian untuk tuning dari kode program ini diperbaiki, bisa saja akan didapatkan hasil sekitar 73 - 74% bahkan bisa melebihi 80%.

## DAFTAR PUSTAKA

- [1] KHAN, ATIF & NAOMIE SALIM, 2014. A Review on Abstractive Summarization Methods. Journal of Theoretical and Applied Information Techonlogy, pp.64-71.
- [2] NALLAPATI, RAMESH., ZHAI, FEIFEI., ZHOU, BOWEN, 2016. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-2017), (pp. 288-290)
- [3] C. SUNITHA., JAYA, DR.A., GANESH, AMAL., 2016, A study on Abstractive Summarization Techniques in Indian Language, Fourth International Conference on Recent

- Trends in Computer Science & Engineering, Elsevier B.V, pp.25-31.
- [4] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).
- [5] JOHNSTON, LINDSAY, 2013, *Data Mining: Concepts, Methodologies, Tools, and Applications*, Vol 1, Information Science Reference, USA.
- [6] DANIEL, GRAUPE, 2013, *Principles of Artificial Neural Network (3rdEdition)*, Word Scientific, Singapore.
- [7] ZHANG, LIN, 2016, *Theory, Methodology, Tools, and Applications for Modeling and Simulation of Complex Systems*, Springer + Business Media, Singapore.
- [8] FAUZI, RAHMAT, 2016, Implementasi Jaringan Syaraf Tiruan dengan Metode Backpropagation terhadap Bibit Tanaman Karet. *Jurnal Education and Development STKIP Tapanuli Selatan*, pp. 1-11.
- [9] K. Cho, dkk. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. arXiv:1406.1078.
- [10] K. Ivanedra, M. Mustikasari. 2018. Implementasi Metode Recurrent Neural Network pada Text Summarization dengan Teknik Abstraktif. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*. 6(4) : 377-382
- [11] Sanjeevi, M. 2018. “Chapter 10.1: DeepNLP — LSTM (Long Short Term Memory) Networks with Math.”. <https://medium.com/deep-mathmachine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235>. Diakses pada 23 Maret jam 02:33 WIB.
- [12] Rozi, I. F., Pramono, S. H, Dahlan, E. A., “Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi,” in *Jurnal EECCIS*, 2012.
- [13] Kouloumpis, E., Wilson, T., Moore, J., “Twitter Sentiment Analysis: The Good the Bad and the OMG!,” in *ICWSM*, 2011.
- [14] Yuliska. dkk. 2020. “Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia”. *IT Journal Research and Development (ITJRD)*. Vol.5, No.1.