

## IMPLEMENTASI *LONG SHORT TERM MEMORY* UNTUK MENDETEKSI UJARAN KEBENCIAN PADA MEDIA SOSIAL DI INDONESIA

Irma Agustina<sup>1</sup>, Suhendro<sup>2</sup>, Sriyanto<sup>3</sup>

<sup>1</sup>Pasca Sarjana, Magister Teknik Informatika, IBI Darmajaya, Bandar Lampung, Indonesia

<sup>2,3</sup>Pasca Sarjana, Magister Teknik Informatika, IBI Darmajaya, Bandar Lampung, Indonesia  
Email: <sup>1</sup>irmaagustina@gmail.com, <sup>2</sup>suhendro@ darmajaya.ac.id, <sup>3</sup>Sriyanto@ darmajaya.ac.id

### Abstrak

Penggunaan media sosial untuk berkomunikasi sesama warga pengguna internet (*netizen*) sudah sangat umum dan menyebar ke seluruh lapisan masyarakat Indonesia. Setiap netizen bebas mengungkapkan ide atau hal yang dirasakannya melalui postingan di media sosial yang kadang tidak terkendali dari segi etika penggunaan bahasa yang dipakai. Pada penelitian ini menggunakan algoritma Long Short-Term Memory (LSTM) pada kasus ujaran kebencian, khususnya yang berkaitan dengan ujaran kebencian dengan pemerintah. Algoritma Long Short-Term Memory merupakan salah satu jenis arsitektur dari Recurrent Neural Network (RNN) yang biasa digunakan pada masalah-masalah yang berkaitan dengan *deep learning*. Penggunaan Word2vec dan metode Bidirectional Long Short Term Memory dengan arsitektur CBOW, dengan epoch 10, learning rate 0,001 dan jumlah neuron 200 pada hidden layer, menghasilkan tingkat akurasi sebesar 94,66%, dengan presisi masing-masing nilai 99,08%, recall 93,74% dan F-measure 96,29%. Sedangkan untuk Bidirectional Long Short Term, Memory dengan tiga layer memiliki akurasi 96,93%. Penambahan satu layer pada BiLSTM meningkat sebesar 2,27%.

**Kata Kunci:** *Long Short Term Memory*, Ujaran Kebencian, Media Sosial, *Deep Learning*, *Neural Network*

### Abstract

*The use of social media to communicate with fellow internet users (netizens) is very common and has spread to all levels of Indonesian society. Every netizen is free to express ideas or things they feel through posts on social media which are sometimes uncontrolled in terms of the ethics of using the language used. In this study, the Long Short-Term Memory (LSTM) algorithm was used in cases of hate speech, especially those related to hate speech with the government. The Long Short-Term Memory algorithm is a type of Recurrent Neural Network (RNN) architecture which is commonly used in problems related to deep learning. The use of Word2vec and the Bidirectional Long Short Term Memory method with CBOW architecture, with epoch 10, learning rate 0.001 and number of neurons 200 in the hidden layer, produces an accuracy level of 94.66%, with a precision value of 99.08%, recall 93.74% and F-measure 96.29%. Meanwhile, for Bidirectional Long Short Term, Memory with three layers has an accuracy of 96.93%. The addition of one layer to BiLSTM increases by 2.27%.*

**Keywords:** *Long Short Term Memory, Hate Speech, Social Media, Deep Learning, Neural Network*

## 1. PENDAHULUAN

Penggunaan media sosial untuk berkomunikasi sesama warga pengguna internet (*netizen*) sudah sangat umum dan menyebar ke seluruh lapisan masyarakat Indonesia. Setiap netizen bebas mengungkapkan ide atau hal yang dirasakannya melalui postingan di media sosial yang kadang tidak terkendali dari segi etika penggunaan bahasa yang dipakai. Berdasarkan identifikasi dari permasalahan di atas, maka penelitian ini merumuskan masalah bagaimana mendeteksi ujaran kebencian dalam media sosial dan merepresentasikannya dengan metode algoritma Long Short Term Memory ?

Deteksi ujaran kebencian adalah tugas klasifikasi teks yang terkait dengan analisis sentimen. Istilah 'ujaran kebencian' secara formal didefinisikan

sebagai 'komunitas apa pun kation yang meremehkan seseorang atau kelompok atas dasar beberapa karakteristik seperti ras, warna kulit, etnis, jenis kelamin, orientasi seksual, kebangsaan, agama, atau karakteristik lain.

Melakukan penelitian untuk mendeteksi berita palsu menggunakan *deep learning*. Data yang diambil berasal dari situs Kaggle.com dengan *dataset fake news* dan *dataset real news* dalam bahasa Inggris. Dalam penelitiannya, dibandingkan hasil dari teknik *machine learning* dengan teknik *deep learning*. Metode yang digunakan adalah *K-Nearest Neighbor*, *Decision Tree*, *Naïve Bayes*, *Random Forest*, *CNN (Convolutional Neural Network)* dan *LSTM (Long Short Term Memory)*. Proses pembentukan vektor yang digunakan adalah teknik *Term Frequency – Inverse Document Frequency (TF-IDF)*. Hasil membuktikan bahwa

teknik *deep learning* dengan model CNN dan LSTM memiliki hasil akurasi yang lebih baik dibandingkan teknik *machine learning* dengan model *K-Nearest Neighbor*, *Decision Tree*, *Naïve Bayes*, dan *Random Forest*. Ananth et al (2019) menyatakan, ketersediaan *dataset* dan literatur untuk mendeteksi berita palsu terbatas. Lingkup masa depan yang disarankan adalah membangun sebuah sistem berdasarkan model yang sudah terbentuk sehingga dapat dimanfaatkan langsung dalam menentukan berita palsu [1].

**Tabel 1. Contoh Data Corpus**

| No | Sumber                      | Contoh Data  |
|----|-----------------------------|--|
| 1  | Data campuran               | Sementara ketiga kelompok kecil itu sedang berusaha menelusuri gerak pasukan Pati, maka para prajurit dan pengawal yang berada di perkemahan tetap bersiaga sepenuhnya untuk menghadapi segala ke?   |
| 2  | Website atau situs berita   | Mungkin bom rakit-an, tapi masih dalam penelitian laboratorium forensik (labfor),” kata Kabid Humas Polda Sulsel, Joko Subroto, Senin (03/07).   |
| 3  | Website dipilih secara acak | Misalnya ditahun ini ada anggaran Program Pemberdayaan Masyarakat Kelurahan (PPMK) Rp540 juta, kita bisa meningkatkan anggaran itu, tetapi tidak juga menguap, dan harus ada pengawasan dan kerjasama dengan pihak yang sukses dalam memberdayakan masyarakat. |

LSTM (*Long Short Term Memory*) merupakan algoritma Deep Learning yang populer dan cocok digunakan untuk membuat prediksi dan klasifikasi yang berhubungan dengan waktu. Algoritma ini bisa dikatakan pengembangan atau salah satu jenis dari algoritma RNN (*Recurrent Neural Network*). Dalam algoritma RNN, output dari langkah terakhir diumpungkan kembali sebagai input pada langkah yang sedang aktif. Namun, algoritma RNN memiliki kekurangan yaitu tidak dapat memprediksi kata yang disimpan dalam memori jangka panjang. LSTM memiliki beberapa gerbang yang memiliki fungsi dan tugasnya masing-masing. Berikut penjelasan singkat mengenai struktur LSTM.[7]

#### 1. Forget Gate

Gerbang pertama dalam LSTM disebut dengan *forget gate*. Mudah-mudahan, gerbang ini bertugas untuk melupakan beberapa informasi yang

tidak relevan dan sudah tidak diperlukan oleh sebuah sistem. Alhasil, LSTM dapat menyajikan kumpulan informasi yang lengkap, tetapi tetap aktual sesuai dengan kebutuhan.

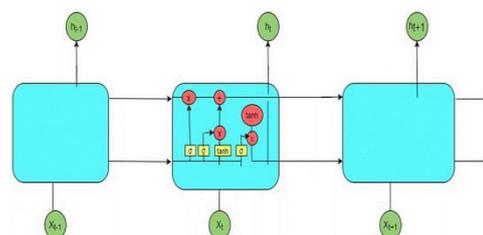
#### 2. Input Gate

Gerbang kedua, yakni *input gate* yang bertugas untuk memasukkan informasi yang berguna untuk mendukung keakuratan data. Tugas *input gate* adalah untuk menambahkan informasi yang sebelumnya telah diseleksi terlebih dahulu melalui gerbang *forget gate*. Gerbang ini tidak dimiliki oleh RNN yang hanya memungkinkan satu *input* data untuk satu *output* data. Dalam *input gate* kemudian dikenal istilah *input modulation gate* yang sering tidak ditulis dalam beberapa ulasan tentang LSTM. Sesuai namanya, *input modulation gate* berfungsi untuk memodulasi informasi yang ada, sehingga dapat mengurangi kecepatan konvergensi dari data *zero-mean*.

#### 3. Output Gate

Terakhir adalah *output gate* yang menjadi gerbang terakhir untuk menghasilkan informasi data yang lengkap dan aktual. Gerbang ini bisa menjadi yang terakhir atas sebuah informasi atau hanya menjadi bagian dari tahap pertama saja, sebelum akhirnya informasi akan diproses lewat *input gate* di sel berikutnya.

Pembelajaran mendalam adalah jenis pembelajaran mesin dan *Artificial Intelligence (AI)* yang meniru cara manusia memperoleh jenis pengetahuan tertentu. Pembelajaran mendalam adalah elemen penting dari ilmu data, yang mencakup statistik dan pemodelan prediktif. Ini sangat bermanfaat bagi ilmuwan data yang bertugas mengumpulkan, menganalisis, dan menafsirkan data dalam jumlah besar; pembelajaran mendalam membuat proses ini lebih cepat dan lebih mudah.[7]



**Gambar 1. Arsitektur Algoritma**

Secara sederhana, cara kerja algoritma LSTM dapat dijabarkan dalam langkah-langkah berikut:

Langkah 1: LSTM memutuskan informasi apa yang harus tetap utuh dan apa yang harus dibuang dari cell state. Lapisan sigmoid bertanggung jawab untuk membuat keputusan ini.

Langkah 2: LSTM menentukan informasi baru apa yang harus disimpan dan menggantikan yang tidak

relevan yang berhasil diidentifikasi pada langkah 1. Fungsi tanh dan sigmoid memainkan peran penting dalam mengidentifikasi informasi yang relevan.

Langkah 3: Output ditentukan dengan bantuan cell state yang sekarang akan menjadi versi yang difilter karena fungsi sigmoid dan tanh yang diterapkan.

Program komputer yang menggunakan pembelajaran mendalam melalui proses yang hampir sama dengan pembelajaran balita untuk mengidentifikasi anjing. Setiap algoritma dalam hierarki menerapkan transformasi nonlinier ke inputnya dan menggunakan apa yang dipelajarinya untuk membuat model statistik sebagai output. Iterasi berlanjut hingga output mencapai tingkat akurasi yang dapat diterima. Jumlah lapisan pemrosesan yang harus dilalui data adalah yang menginspirasi dalam label.[4]

Rahutomo et al (2019) melakukan eksperimen menggunakan *Naïve Bayes* untuk mendeteksi hoaks berbahasa Indonesia. Jumlah *dataset* yang digunakan adalah 600 berita yang terdiri dari berita fakta dan berita palsu berbasis teks. Metode yang digunakan terdiri dari *preprocessing*, *manual voting tagging* untuk pelabelan berita fakta atau berita palsu, implementasi model *Naïve Bayes Classifier*, kemudian melakukan pengujian statis dan dinamis. Hasil evaluasi dilakukan menggunakan 3 parameter yaitu *accuracy*, *precision* dan *recall*. Akurasi yang dihasilkan dengan pengujian statis sebesar 82.6%, sedangkan akurasi yang dihasilkan dengan pengujian dinamis sebesar 68.33%[9]

Putri et al (2019) melakukan analisis dan deteksi berita palsu menggunakan berita dalam bahasa Indonesia. Data yang digunakan berjumlah 251 artikel berita yang terdiri dari 151 artikel berita fakta dan 100 artikel berita palsu. Dalam penelitiannya, Putri et al (2019) menggunakan *machine learning* dengan beberapa metode seperti *text preprocessing* dan *feature extraction*, kemudian membandingkan 5 model algoritma *machine learning*, yaitu *Multilayer Perceptron*, *SVM (Support Vector Machine)*, *Naïve Bayes*, *Random Forest* dan *Decision Tree* [20]

## 2. METODOLOGI PENELITIAN

### 2.1. Jenis Penelitian

Jenis penelitian dalam tesis ini adalah penelitian eksperimen. Penelitian eksperimen adalah Penelitian eksperimen adalah penelitian yang dilakukan dengan pendekatan saintifik dengan menggunakan dua set variabel. Set pertama bertindak sebagai konstanta, yang dapat digunakan untuk mengukur perbedaan dari set kedua. Metode penelitian kuantitatif, misalnya, bersifat eksperimental.

### 2.2. Sumber Data Penelitian

Dalam penelitian ini terdapat dua sumber data, yaitu sumber data primer dan sumber data sekunder.

#### 2.2.1. Sumber Data Primer

Sumber data primer dalam penelitian ini ialah *dataset* kalimat normal maupun ujaran kebencian berasal dari komentar di media sosial Twitter yang dapat diakses oleh semua pihak, berupa teks yang diambil melalui penelusuran secara *online* pada kolom komentar Twitter antara tanggal 21-22 Desember 2022.

Tabel 2. Contoh Ujaran Kebencian

| No | Alamat                   | Kalimat ujarannya   | Tanggal                  |
|----|--------------------------|---|--------------------------|
| 1  | @worksfeed (netral)      | petugas imigrasi tu emang jutek ya? gua cuman nanya doang padahal, nanyanya juga baik2. bikin emosi aja malem2  | 21 Des 2022<br>09.40 PM  |
| 2  | @ruhutsitompul (Negatif) | Ha ha ha yg gini mau jadi Presiden RI ?, yg ta'unya hanya ganti2 nama Rumah Sakit jadi Rumah Sehat ganti nama2 jalan eh Rumah Makan jadi Rumah Kenyang dasar kadrun | 22 Des 2022<br>5.18 PM   |
| 3  | @ainunrozi (merendahkan) | kadang pengeeeeeen banget nolol-nololin orang depan mukanya   | 22 Des 2022,<br>12.09 PM |

#### 2.2.2. Pelabelan Twitter API

Twitter API memungkinkan akses terprogram ke Twitter dengan cara yang unik dan canggih. Manfaatkan elemen inti Twitter seperti: Tweet, Pesan Langsung, Ruang Daftar pengguna, dan lainnya.

#### 2.2.3. Sumber Data Sekunder

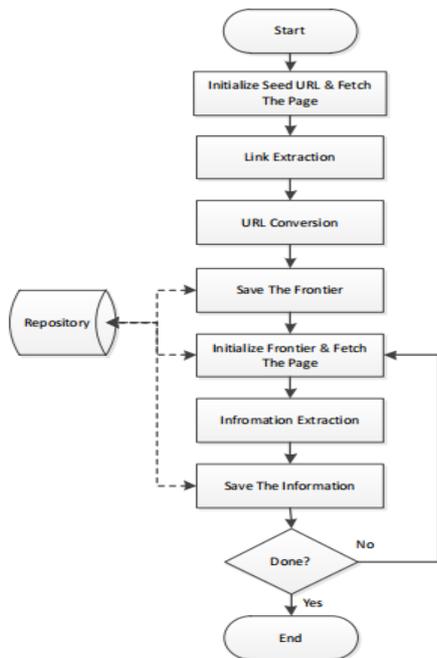
Sementara sumber data sekunder dalam penelitian ini ialah beragam pustaka referensi dan artikel ilmiah dalam jurnal berkaitan dengan topik penelitian.

### 2.3. Alat Pengumpulan Data

#### A. Web Crawler

Web Crawler adalah program yang menelusuri World Wide Web dengan cara yang metodis, otomatis dan teratur. Istilah lain untuk web crawler

adalah ant, automatic indexer, bots, web spiders atau web robots. Web Crawler adalah salah satu jenis bot atau agen perangkat lunak. Secara umum, proses crawling dimulai dengan list URL yang akan dikunjungi, disebut seeds. Kemudian web crawler akan mengunjungi URL tersebut satu per satu. Setiap page URL yang dikunjungi akan diidentifikasi apakah ada hyperlink di dalamnya.



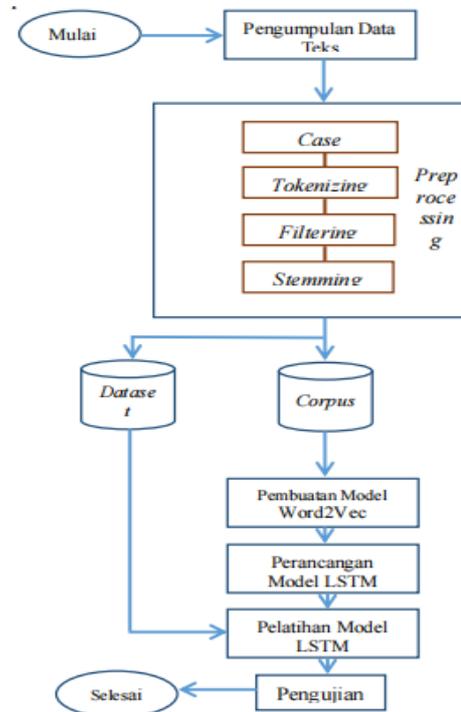
Gambar 2. Data Web Crawler

jika ada maka akan ditambahkan ke dalam list URL yang akan dikunjungi. Ini disebut crawl frontier.

#### 2.4. Tahapan dalam Penelitian

Secara umum, tahapan penelitian ini diawali dengan studi literatur, dilanjutkan dengan pengumpulan data untuk membuat *corpus* dan *dataset* yang berasal dari data *corpus* dari Twitter. *Data preprocessing* (pra-pemrosesan data) yang dilakukan terhadap data teks (*corpus* dan *dataset*) adalah pembersihan *stopwords*, *stemming*, dan menghilangkan karakter-karakter yang tidak diperlukan.

Hasil dari pra-pemrosesan dikumpulkan ke dalam sebuah *file* teks. Selanjutnya, tahapan pembuatan model *Word2Vec*. Teks hasil prapemrosesan diubah ke dalam bentuk vektor agar dapat dibaca oleh algoritma Long Short-Term Memory (LSTM). Dilanjutkan dengan merancang model *Long-Short Term Memory* yang sesuai dengan *corpus* dan *dataset* agar model yang dirancang dapat mendeteksi penggunaan ujaran kebencian dengan baik, model dilatih dengan menggunakan *dataset* yang berisi kumpulan kalimat mengandung ujaran kebencian, dan diakhiri dengan pengujian model.



Gambar 3. Flowchart

Sebelum dilakukan pembuatan model *Word2Vec*, dilakukan pra-pemrosesan terlebih dahulu menggunakan *library Python* yaitu *gensim* [9] pada *corpus* dan *dataset*. *Library gensim* mengandung fungsi untuk melakukan prapemrosesan teks yang meliputi:

##### 1. Case Folding

Proses *Case Folding* ini bertujuan untuk mengubah huruf dalam teks menjadi huruf standar (huruf kecil). Data yang diterima pada *case folding* hanya huruf 'a' sampai 'z', karakter selain huruf-huruf tersebut dihilangkan dan hanya dianggap delimiter.

##### 2. Tokenizing

Setelah melakukan *case folding*, proses selanjutnya adalah data teks dipecah per kata pada tahapan *tokenizing*.

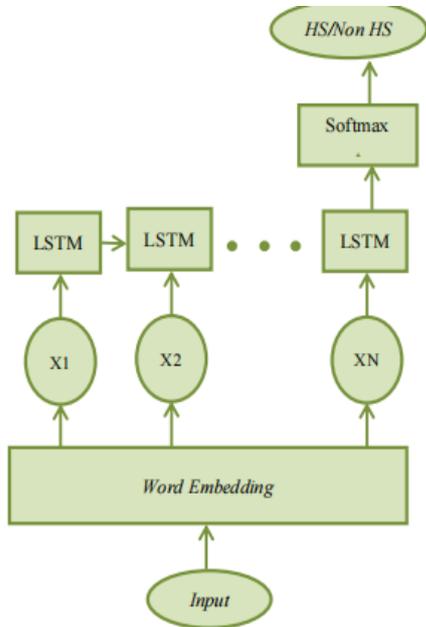
##### 3. Filtering

Pada tahap *filtering* dilakukan proses penghapusan kata yang tidak diperlukan atau diluar objek penelitian (*stopwords*). Beberapa kata *stopwords* pada penelitian ini diantaranya adalah 'maka', 'akan', 'yang', 'untuk', 'dan', 'juga', 'dari', 'di', 'ya', serta 'kan'.

##### 4. Stemming

Setelah proses *filtering* dilakukan, proses yang terakhir adalah *stemming*, dimana kata yang ada pada data ditransformasi ke kata dasarnya. Dalam perancangan *Word2Vec*, digunakan *library gensim*.

Library *gensim* diperlukan untuk melakukan *import* fungsi *Word2Vec*, dimana fungsi ini digunakan untuk mengubah teks pada "*corpus.txt*" dari kata menjadi vektor untuk dapat dijalankan pada algoritma LSTM dan hasil dari *Word2Vec* disimpan ke *model\_word2vec\_300.model*.



Gambar 4. Algoritma LSTM

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengujian Arsitektur Word2vec

Pengujian pertama adalah menguji arsitektur *word2vec* yang digunakan yaitu tipe *Continuous Bag-Of-Word (CBOW)* dan *skip-gram*. Sedangkan untuk membandingkan dengan metode lain selain proses *word2vec*, akan diuji salah satu metode *hot encoding*. Pengujian ini bertujuan untuk menentukan jenis *word2vec* terbaik dan satu model *hot encoding*.

#### 3.2. Pengujian Jumlah Neuron Terhadap Metode Bi\_LSTM

Pada pengujian ini diberi nilai masa 10 dan L2 sebesar 0,001. Berdasarkan tabel 4 terlihat bahwa jumlah neuron hingga 50 nilai akurasi yang diperoleh tidak dapat memberikan peningkatan akurasi yang signifikan, jumlah neuron terus bertambah hingga 300 neuron, ketika jumlah neuron 200 diperoleh nilai akurasi sebesar 94,66 %, pada jumlah neuron 220 menjadi 300 terjadi penurunan nilai akurasi sebesar 24,47%, jumlah neuron 300 mendapatkan nilai akurasi sebesar 70,19%. Dari hasil percobaan ini didapatkan nilai akurasi tertinggi saat jumlah neuron 200 yaitu 94,66%. Semakin banyak jumlah neuron maka semakin lama waktu komputasi yang dibutuhkan. Di sisi lain, jumlah neuron yang banyak tidak

menjamin dapat meningkatkan nilai pengukuran akurasi, presisi, recall, dan F yang signifikan.

#### 3.3. Pengujian Jumlah Epoch Terhadap Metode Bi\_LSTM

Pada pengujian sebelumnya telah ditentukan jumlah neuron berdasarkan hasil pengujian yang menunjukkan nilai terbaik. Pada penelitian ini jumlah epoch yang diuji adalah 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 sedangkan jumlah neuron adalah 200 karena memiliki akurasi tertinggi berdasarkan pengujian sebelumnya dan Nilai L2 sebesar 0,001. Pada Tabel 4.5. terlihat bahwa jumlah epoch 100 memberikan hasil dengan akurasi terbaik.

| Number of Epoch | Time (minutes) | Accuracy (%) | Loss |
|-----------------|----------------|--------------|------|
| 5               | 04:02          | 63,89        | 0,65 |
| 10              | 08:47          | 64,38        | 0,61 |
| 20              | 10:05          | 86,72        | 0,43 |
| 30              | 13:09          | 87,63        | 0,3  |
| 40              | 15:23          | 86,83        | 0,25 |
| 50              | 17:15          | 92,97        | 0,2  |
| 60              | 17:55          | 90,24        | 0,16 |
| 70              | 18:20          | 91,60        | 0,22 |
| 80              | 18:57          | 96,63        | 0,11 |
| 90              | 19:10          | 97,87        | 0,08 |
| 100             | 19:55          | 98,10        | 0,07 |

#### 3.4. Pengujian Regularisasi L2

Pengujian sebelumnya telah mendapatkan arsitektur *word2vec*, jumlah neuron terbaik, dan jumlah epoch. Terakhir, tentukan nilai regularisasi L2 yang optimal. Nilai L2 yang diuji adalah 0,1, 0,01 dan 0,001. Hasil pengujian dapat dilihat pada Tabel 3

#### 3.5. Pengujian Kinerja Bi-LSTM

Berdasarkan pengujian sebelumnya didapatkan arsitektur *Word2vec* yaitu *CBOW*, jumlah neuron, jumlah epoch, dan nilai Regularisasi L2 terbaik. Hasil terbaik dari pengujian sebelumnya akan digunakan untuk menguji performansi metode *Bi-LSTM* dan akan dibandingkan dengan metode LSTM yaitu lapisan *Bi-LSTM 3*. Hasil pengujian ditunjukkan pada Tabel 8 hasil pengujian *BiLSTM*.

Tabel 3. Hasil Perbandingan Pengujian BiLSTM

|                 | Accuracy (%) | Precision (%) | Recall (%) | F-measure (%) |
|-----------------|--------------|---------------|------------|---------------|
| Bi-LSTM         | 94,66        | 99,08         | 93,74      | 96,29         |
| Bi-LSTM 3 Layer | 96,93        | 99,74         | 94,61      | 97,02         |
| LSTM            | 88,78        | 96,92         | 91,74      | 94,16         |

## 4. SIMPULAN

### 4.1. Simpulan

Berdasarkan pembahasan di atas dapat disimpulkan bahwa :

1. Penggunaan Word2vec dan metode Bidirectional Long Short Term Memory dengan arsitektur CBOW, dengan epoch 10, learning rate 0,001 dan jumlah neuron 200 pada hidden layer, menghasilkan tingkat akurasi sebesar 94,66%, dengan presisi masing-masing nilai 99,08%, recall 93,74% dan F-measure 96,29%.
2. Sedangkan untuk Bidirectional Long Short Term, Memory dengan tiga layer memiliki akurasi 96,93%. Penambahan satu layer pada BiLSTM meningkat sebesar 2,27%.

## 5. REFERENCES

- [1]. Ananth, S., Radha, D. K., Prema, S., D., & Nirajan, K. (2019). Fake News Detection using Convolution Neural Network in Deep Learning. *International Journal Of Innovative Research In Computer And Communication Engineering*, 7(1).
- [2]. M. S. Jahan and M. Oussalah, "A systematic review of Hate Speech automatic detection using Natural Language Processing," 2021, [Online]. Available: <http://arxiv.org/abs/2106.00742>
- [3]. M. W. P. Aldi, Jondri, and A. Aditsania, "Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin," *J. Inform.*, vol. 5, No. 2, p. 3548, 2018, [Online]. Available: <http://openlibrarypublications.telkomniversity.ac.id>
- [4]. A. M. S. Al-Hamzi, A. Gougui, Y. Sari Amalia, and T. Suhardijanto, "Corpus Linguistics and Corpus-Based Research and Its Implication in Applied Linguistics: A Systematic Review," *Parol. J. Linguist. Educ.*, vol. 10, no. 2, pp. 176–181, 2020, doi: 10.14710/parole.v10i2.176-181.
- [5]. M. H. Ataie, "Basic Implementation of sentiment analysis using BERT .," no. February, 2022.
- [6]. V. Cotik *et al.*, "A study of Hate Speech in Social Media during the COVID-19 outbreak," *ACL 2020 Work. NLP-COVID*, no. 1, p. 6, 2020, [Online]. Available: <https://openreview.net/forum?id=01eOESDhbSW>
- [7]. R. Dwi, W. Santosa, M. A. Bijaksana, and A. Romadhony, "Implementasi Algoritma Long Short-Term Memory ( LSTM ) untuk Mendeteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia," *J. Tugas Akhir Fak. Inform.*, vol. 8, no. 1, pp. 691–702, 2021.
- [8]. A. Geet *et al.*, "Classification of Hate Speech Using Deep Neural Networks To cite this version : HAL Id : hal-03101938 Classification of Hate Speech Using Deep Neural Networks," 2021.
- [9]. P. Kapil, A. Ekbal, and D. Das, "Investigating Deep Learning Approaches for Hate Speech Detection in Social Media," *arXiv*, pp. 1–12, 2020.
- [10]. A. Khumaidi, R. Raafi'udin, and I. P. Solihin, "Pengujian Algoritma Long Short Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung," *J. Telemat.*, vol. 15, no. 1, pp. 13–18, 2020, [Online]. Available: <https://journal.itb.ac.id/telematika/article/view/340>
- [11]. A. A. Kurniawan and M. Mustikasari, "Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia," *J. Inform. Univ. Pamulang*, vol. 5, no. 4, p. 544, 2021, doi: 10.32493/informatika.v5i4.6760.
- [12]. Mohiyaddeen and Dr. Shifaula Siddiqui, "Automatic Hate Speech Detection: A Literature Review," *Int. J. Eng. Manag. Res.*, vol. 11, no. 2, pp. 116–121, 2021, doi: 10.31033/ijemr.11.2.17.
- [13]. Mohiyaddeen and Dr. Shifaula Siddiqui, "Automatic Hate Speech Detection: A Literature Review," *Int. J. Eng. Manag. Res.*, vol. 11, no. 2, pp. 116–121, 2021, doi: 10.31033/ijemr.11.2.17.
- [14]. M. A. Nurrohmat and A. SN, "Sentiment Analysis of Novel Review Using Long Short-Term Memory Method," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 3, p. 209, 2019, doi: 10.22146/ijccs.41236.
- [15]. M. A. Paz, J. Montero-Díaz, and A. Moreno-Delgado, "Hate Speech: A Systematized Review," *SAGE Open*, vol. 10, no. 4, 2020, doi: 10.1177/2158244020973022.
- [16]. E. Perdana Prasetya, A. Dewi Ekawati, D. Sapta Nugraha, A. Marzuq, and T. Saputri Darlis, "Corpus Linguistics, Language Corpora and Language Teaching," *English J.*, vol. 14, no. 2, p. 75, 2020, doi: 10.32832/english.v14i2.3845.
- [17]. E. D. Pratama, "Implementasi Model Long-Short Term Memory (LSTM) pada Klasifikasi Teks Data SMS Spam Berbahasa Indonesia," *J. Mach. Learn. Comput. Intell.*, vol. 1, no. 2, p. 2022, 2022.
- [18]. P. A. Qori, D. S. Oktafani, and I. Kharisudin, "Analisis Peramalan dengan Long Short Term Memory pada Data Kasus Covid-19 di Provinsi Jawa Tengah," *Prism. Pros. Semin. Nas. Mat.*, vol. 5, pp. 752–758, 2022.
- [19]. F. D. Souza and J. B. de O. e. S. Filho, "BERT for Sentiment Analysis: Pre-trained and Fine-Tuned Alternatives," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13208 LNAI, pp. 209–218, 2022, doi: 10.1007/978-3-030-98305-5\_20.
- [20]. A. S. Talita and A. Wiguna, "Implementasi Algoritma Long Short-Term Memory (LSTM) Untuk Mendeteksi Ujaran Kebencian (Hate Speech) Pada Kasus Pilpres 2019," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 1, pp. 37–44, 2019, doi: 10.30812/matrik.v19i1.495.
- [21]. L. Wiranda and M. Sadikin, "Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma," *J. Nas. Pendidik. Tek. Inform. JANAPATI*, vol. 8, no. 3, pp. 184–196, 2019, [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/janapati/article/view/19139>
- [22]. S. Zahara, Sugianto, and M. Bahril Ilmiddafiq, "Prediksi Indeks Harga Konsumen

- Menggunakan Metode Long Short Term Memory (LSTM) Berbasis Cloud Computing,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 3, pp. 357–363, 2019, doi: 10.29207/resti.v3i3.1086.
- [23] Z. Zhang, D. Robinson, and J. Tepper, “Hate Speech Detection Using a Convolution-LSTM Based Deep Neural Network,” *Proc. ACM Web Conf. (WWW 2018)*, pp. 1–10, 2018, [Online]. Available: [https://doi.org/10.475/123\\_4](https://doi.org/10.475/123_4)