

Analisis Sentimen menggunakan Recurrent Neural Network Terkait Isu Anies Baswedan Sebagai Calon Presiden 2024

1st Salman Farisi Setya Hadi
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
farissh@student.telkomuniversity.
ac.id

2nd Jondri
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
jondri@telkomuniversity.ac.id

3rd Kemas Muslim Lhaksana
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
kemasmuslim@telkomuniversity.a
c.id

Sosial media twitter saat ini sudah menjadi wadah utama untuk menyalurkan berita dari suatu tempat ke tempat yang lain. Dengan adanya sosial media digital, masyarakat dapat dengan mudah mengetahui berita terkini ataupun memberi opini tentang isu yang sedang hangat. Isu yang ada bisa berasal dari beberapa bidang, seperti contoh dalam bidang politik. Salah satu isu hangat pada bidang politik adalah isu tentang Bapak Anies Baswedan sebagai calon presiden 2024. Perbincangan tersebut akan menghadirkan banyak sentimen positif maupun negatif, maka dari itu diperlukan sistem klasifikasi sentimen yang efektif. Metode yang digunakan pada penelitian ini adalah recurrent neural network yang dikombinasikan dengan word embedding word2vec. Hasil yang dihasilkan dari penelitian ini yaitu nilai akurasi sebesar 86.27%, nilai precision sebesar 88.24% dan nilai recall sebesar 84.91%.

Kata kunci - sentimen, recurrent neural network, word2vec, word embedding

I. PENDAHULUAN

A. Latar Belakang

Sosial media digital saat ini sudah menjadi wadah utama untuk menyalurkan berita dari suatu tempat ke tempat yang lain. Dengan adanya sosial media digital, masyarakat dapat dengan mudah mengetahui berita terkini ataupun memberi opini tentang isu yang sedang hangat dalam berbagai bidang, seperti contoh dalam bidang politik.

Salah satu isu hangat dalam bidang politik Indonesia pada saat ini yaitu pemilihan calon kandidat presiden 2024. Semua jenis media massa secara serentak membicarakan hal tersebut, mulai dari saluran televisi, radio, bahkan sosial media digital seperti twitter ramai memperbincangkan hal ini. Salah satu calon kandidat yang banyak diperbincangkan masyarakat adalah sosok gubernur jakarta Anies Baswedan [1].

Dalam sosial media Twitter, tagar dukung Anies Baswedan Presiden 2024 sempat menjadi *trending topics* [2]. Banyak masyarakat yang antusias untuk berdiskusi tentang kinerja dari bapak Anies Baswedan dan memunculkan spekulasi positif dan negatif, tetapi dengan banyaknya pengguna twitter di Indonesia yang mencatat sebanyak 18,45 juta

Abstract-Digital social media has now become the main medium for distributing news from one place to another. With the existence of digital social media, people can easily find out the latest news or give opinions on issues that are currently hot in various fields, such as for example in the political field. One of the hot issues in the political field is the issue of Mr. Anies Baswedan as a presidential candidate for 2024. This discussion will present many positive and negative sentiments, therefore an effective sentiment classification system is needed. The method used in this study is a recurrent neural network combined with word embedding word2vec. The results of this study are the accuracy value of 86.27%, the precision value of 88.24% and the recall value of 84.91%.

Keyword- Sentiment, Recurrent Neural Network, Word2Vec, Word Embedding

pengguna [3], melakukan klasifikasi sentimen pada isu tersebut sulit jika dilakukan secara manual.

Penelitian terdahulu digunakan untuk membangun sistem klasifikasi pada penelitian ini. Pada penelitian [4], peneliti melakukan analisis sentimen terhadap ulasan film yang bersumber dari IMDB. Sebanyak 50.000 data dengan distribusi dataset seimbang (25.000 berlabel positif dan 25.000 berlabel negatif). Metode yang digunakan pada penelitian tersebut adalah *recurrent neural network* (RNN). Hasil yang didapat dari penelitian tersebut adalah tingkat akurasi sebesar 87.42%, tingkat *recall* sebesar 87.17%, tingkat *precision* sebesar 87.53%, dan tingkat *f-measure* sebesar 87.34%.

Maka dari itu, penulis mengusulkan untuk membuat sebuah sistem klasifikasi untuk melakukan analisis sentimen terhadap isu Anies Baswedan sebagai calon kandidat presiden 2024. Metode yang akan digunakan pada penelitian ini yaitu *Recurrent Neural Network* (RNN). Metode RNN digunakan untuk klasifikasi dikarenakan RNN mampu mengolah data yang bersifat sekuensial [5]. Selain penggunaan metode klasifikasi RNN, penelitian ini juga akan menggunakan metode ekspansi fitur *Word2Vec* untuk membuat metode klasifikasi RNN

menjadi lebih teroptimasi. *Word2vec* dipilih sebagai fitur ekspansi pada penelitian ini karena berdasarkan penelitian [6], metode fitur ekspansi tersebut dapat mengenali korelasi gramatikal antar kata, sehingga membuat metode klasifikasi tersebut lebih optimal.

B. Rumusan Masalah

Adapun rumusan masalah dari latar belakang yaitu:

1. Bagaimana membangun sistem analisis sentimen pada isu Anies Baswedan sebagai calon presiden 2024 menggunakan *recurrent neural network* dengan ekstraksi fitur *word2vec*?
2. Bagaimana performansi sistem klasifikasi *recurrent neural network* pada pengujian analisis sentimen dalam isu Anies Baswedan sebagai calon presiden 2024?

C. Tujuan

Adapun tujuan dari penelitian ini yaitu:

1. Membuat sebuah sistem analisis sentimen pada isu Anies Baswedan sebagai calon presiden 2024 menggunakan *recurrent neural network* dengan ekstraksi fitur *word2vec*.
2. Mengetahui performansi sistem klasifikasi *Recurrent Neural Network* pada pengujian analisis sentimen dengan topik isu Anies Baswedan sebagai calon presiden 2024.

D. Batasan Masalah

Adapun batasan dari penelitian ini yaitu:

1. Melakukan analisis sentimen menggunakan *recurrent neural network* (RNN) dengan *word embedding word2vec*.
2. Data yang digunakan berupa cuitan berbahasa Indonesia dengan kata kunci 'anies baswedan' yang berasal dari *twitter*.

II. KAJIAN TEORI

A. Penelitian Terkait

Sejumlah penelitian tentang topik terkait sentimen analisis telah dilakukan oleh banyak pihak. Pada penelitian [7], peneliti melakukan penelitian analisis sentimen terhadap *tweet* dengan topik Covid-19 dengan menggunakan *Word Embedding Word2vec* dan *Long Short-Term Memory* (LSTM). Tujuan dari penelitian ini adalah untuk mencari kebijakan saat Covid-19 mendapat sentimen positif atau negatif. Hasil yang didapat pada penelitian tersebut menunjukkan bahwa penelitian tersebut

mendapatkan tingkat akurasi sebesar 81%, precision sebesar 80%, recall sebesar 80%, dan nilai F1-Measure sebesar 81%.

Pada penelitian [8], peneliti melakukan analisis sentimen terhadap *movie review* dengan menggunakan *word embedding Word2Vec* dan LSTM. Dengan dataset sebanyak 25.000 data dengan distribusi data 12.500 data positif dan 12.500 data negatif. Hasil yang didapat dari penelitian tersebut menunjukkan bahwa tingkat akurasi tertinggi yang didapat sebesar 88.17% dengan menggunakan dimensi *word vector* sebesar 100, dan tingkat akurasi terendah 85.86% dengan menggunakan dimensi *word vector* sebesar 500.

Pada penelitian [9], peneliti melakukan analisis sentimen terhadap bakal calon presiden 2024 yaitu Ganjar Pranowo, Anies Baswedan, Prabowo Subianto, dan Ridwan Kamil. Hasil yang didapat dari penelitian ini adalah tingkat akurasi untuk dataset tentang Ganjar Pranowo sebanyak 274 sebesar 73.68% pada *fold* ke-7. Tingkat akurasi pada dataset tentang Anies Baswedan sebanyak 120 data sebesar 71.43% pada *fold* ke-5. Tingkat akurasi pada dataset tentang Prabowo Subianto sebanyak 72 Data sebesar 60% pada *fold* ke-1. Tingkat akurasi pada dataset tentang Ridwan Kamil sebanyak 67 data sebesar 62.5%.

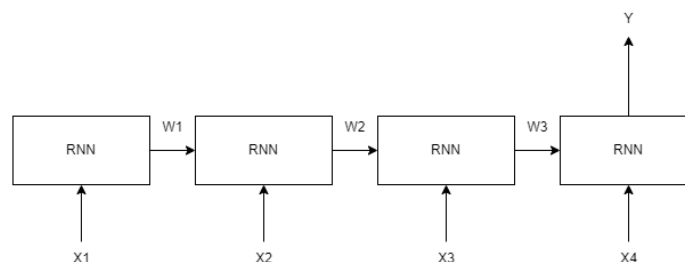
B. Dasar Teori

1. Analisis Sentimen

Analisis sentimen atau bisa disebut sebagai penambangan opini adalah sebuah bidang studi yang bertujuan untuk menganalisa opini dari masyarakat, seperti contoh analisa emosi tentang sebuah produk, organisasi, individu, layanan, maupun topik [10]. Pada penelitian ini, opini masyarakat digunakan untuk menilai tanggapan masyarakat terhadap isu tentang Anies Baswedan sebagai calon presiden 2024.

2. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah salah satu metode klasifikasi dalam kelompok *deep learning*. Pada dasarnya, RNN bekerja dengan cara menganalisa melalui semua data dan parameternya digunakan pada saat setiap kali dibagikan. RNN juga menyimpan salinan dari data sebelumnya untuk komputasi selanjutnya [11]. Berikut ini adalah gambar dari arsitektur RNN:



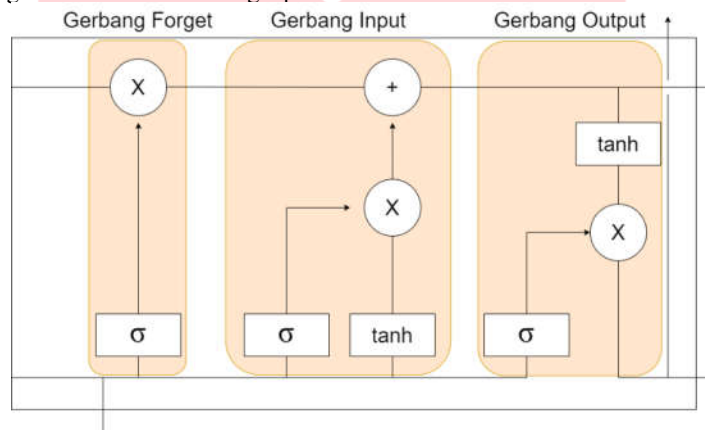
GAMBAR 1
ARSITEKTUR RNN MANY-TO-ONE

Gambar diatas adalah representasi dari arsitektur RNN. RNN memiliki 3 layer utama sebagai arsitekturnya. Layer pertama adalah *input layer* yang berfungsi untuk menangkap data *input*. Layer kedua adalah *recurrent layer*. *Recurrent layer* berfungsi untuk menangkap hasil komputasi yang terdapat diantara layer pertama dan layer ketiga. Selain itu, *recurrent layer* berfungsi untuk mengingat hasil komputasi yang telah dilakukan untuk diolah kembali ketika terdapat data baru dari *input layer*. Layer ketiga adalah *output layer*. *Output layer* berfungsi untuk menangkap hasil komputasi dari *recurrent layer*. Tipe RNN yang digunakan pada penelitian ini yaitu *many-to-one* yang berarti RNN menangkap

beberapa sekuens yang akan menghasilkan satu *output*, seperti gambar diatas.

3. Long Short-term Memory (LSTM)

Long Short-term Memory (LSTM) adalah salah satu jenis arsitektur jaringan saraf tiruan yang merupakan model modifikasi dari *recurrent neural network* (RNN). LSTM hadir untuk mengatasi masalah hilangnya efektifitas gradien pada RNN [12]. LSTM dan RNN mempunyai struktur jaringan yang serupa, yang membedakan adalah pada LSTM terdapat gerbang tertentu yang berfungsi sebagai pengaturan aliran informasi.



GAMBAR 2
ARSITEKTUR LSTM [13]

a. Forget Gate

Forget gate dalam LSTM berfungsi untuk menghilangkan informasi apapun yang tidak

dibutuhkan atau kurang memiliki makna. Proses menghilangkan informasi tersebut dilakukan oleh fungsi *sigmoid*. Adapun rumus perhitungan nilai pada *forget gate* adalah sebagai berikut :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_i) \quad (1)$$

Dengan keterangan:

f_t = nilai dari *forget gate*

σ = fungsi *sigmoid*

W_f = bobot pada waktu ke t

h_{t-1} = nilai output pada waktu ke $t-1$

x_t = nilai input pada waktu ke t

b_i = nilai bias pada *forget gate*

b. Input Gate

Setelah informasi diolah dalam *forget gate*, selanjutnya informasi akan diolah di dalam *input gate*. *Input gate* berfungsi untuk memperbarui informasi yang telah di dapat menggunakan fungsi aktivasi *sigmoid*. *Input gate* juga akan membentuk vektor baru menggunakan fungsi aktivasi *tanh*. Berikut ini adalah rumus perhitungan pada *input gate*:

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

Dengan keterangan:

i_t = nilai dari *input gate*

c. Cell State

Fungsi *cell state* pada arsitektur LSTM adalah untuk memperbarui nilai *cell state* lama hasil pengolahan pada *input gate*. Untuk melakukan

pembaruan terhadap nilai *cell state* lama, rumus yang digunakan yaitu:

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (3)$$

Dengan keterangan:

c_t = nilai dari *cell state* pada waktu ke t

c_{t-1} = nilai dari *cell state* pada waktu ke $t-1$

\tilde{c}_t = nilai dari *candidate cell state* pada waktu ke t

d. Output Gate

Gerbang terakhir pada arsitektur LSTM yaitu *output gate*. Gerbang ini memiliki fungsi untuk menghasilkan nilai akhir fungsi aktivasi *sigmoid* dan *tanh*. Adapun rumusnya adalah sebagai berikut:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$h_t = o_t * \tanh(c_t) \quad (5)$$

Dengan keterangan:

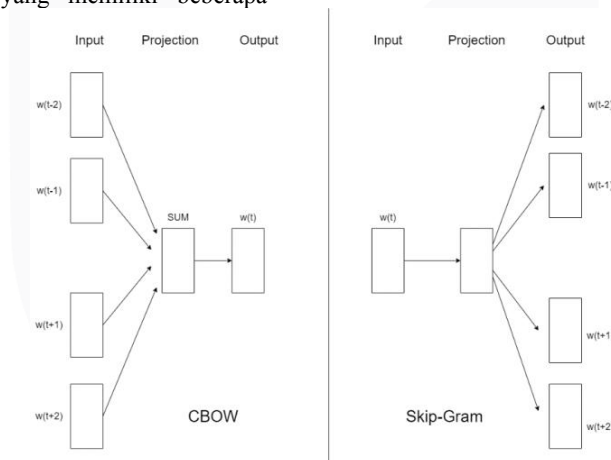
o_t = nilai dari *output gate* aktivasi *sigmoid* pada waktu ke t

h_t = nilai dari *output gate* aktivasi *tanh* pada waktu ke t

4. Word2Vec

Word2vec adalah salah satu metode dalam *word embedding* yang dapat mengubah sebuah kata menjadi besaran *vector*. *Word2vec* ini bekerja dalam struktur jaringan saraf yang memiliki beberapa

lapisan yaitu lapisan masukan, lapisan keluaran, dan lapisan tersembunyi. *Word2vec* ini mempunyai 2 arsitektur yang berbeda, yaitu *continuous bag-of-word* (CBOW) dan *skip gram* [14]. Kedua arsitektur ini mempunyai perbedaan pada distribusinya. Pada arsitektur CBOW, distribusi konteksnya dikombinasikan kepada kata prediksi, sedangkan pada arsitektur *skip gram*, distribusi konteksnya merepresentasikan dari kata masukan yang digunakan untuk memprediksi sebuah kata.

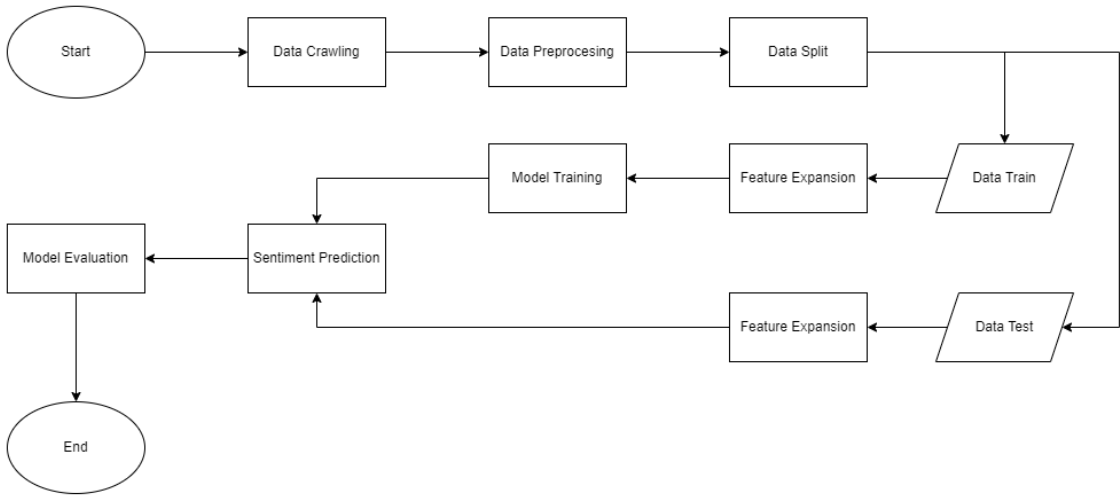


GAMBAR 3
ARSITEKTUR *WORD2VEC* [15]

III. METODE

Pada penelitian ini, terdapat beberapa tahapan yang akan dilakukan, diantaranya adalah pengumpulan data, preprocessing data, ekspansi

fitur, dan klasifikasi. Berikut ini adalah perancangan sistem yang akan dibuat pada penelitian ini :



GAMBAR 4
RANCANGAN SISTEM

A. Data Crawling

Pada penelitian ini, data yang akan digunakan berasal dari cuitan dari aplikasi sosial media *twitter*. Pengambilan data dilakukan dengan cara melakukan *scrapping* menggunakan *library sncrape*. *Sncrape* adalah salah satu *library* dari bahasa pemrograman *python* yang digunakan untuk melakukan *scrapping tweet*. Rentan waktu untuk data yang diambil dari tanggal 1 Juni 2022 sampai 30 Juni 2022.

B. Data Preprocessing

Pada tahap data *preprocessing*, ada beberapa hal yang harus dilakukan yaitu *case folding*, *special word removal*, *punctuation removal*, *stop word removal*, *tokenization*, dan *stemming*.

1. Case Folding

Case folding adalah tahapan untuk mengubah semua awalan huruf kapital menjadi non kapital.

TABEL 1
CASE FOLDING

Sebelum	Sesudah
@ruhutsitompul Kita harus akui bahwa Anies Baswedan Goodbener Indonesia Gubernur rasa presiden The next president 2024.	@ruhutsitompul kita harus akui bahwa anies baswedan goodbener indonesia gubernur rasa presiden the next president 2024.

2. Special Word Removal

Special word removal bertujuan untuk menghilangkan angka, tagar, *mention*, dan memperbaiki *spacing* dalam *tweet*.

TABEL 2
SPECIAL WORD REMOVAL

Sebelum	Sesudah

	Negatif	False Positive (FP)	True Negative (TN)
--	---------	------------------------	--------------------

Adapun hal penunjang yang akan dijadikan untuk evaluasi adalah akurasi, precision, dan recall yang dikalkulasikan berdasarkan matriks klasifikasi diatas.

1. Akurasi

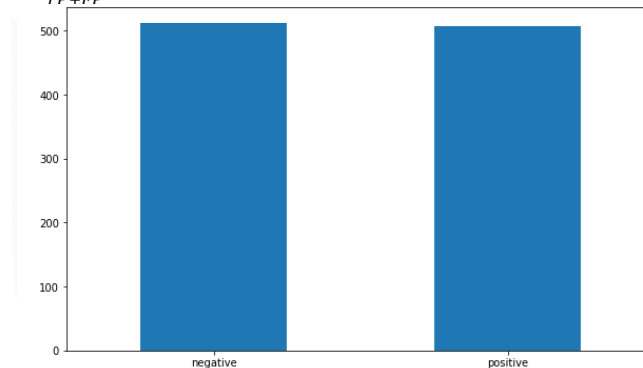
Akurasi adalah sebuah matriks untuk model klasifikasi yang mengukur sejumlah prediksi yang tepat berdasarkan keseluruhan data.

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

2. Precision

Precision adalah sebuah matriks untuk model klasifikasi yang mengukur ketepatan prediksi dari model yang sudah dibuat.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$



GAMBAR 4
DISTRIBUSI *TWEET* SENTIMEN POSITIF DAN NEGATIF

Dari 1019 dataset yang didapat menghasilkan 512 kelas negatif dan 507 kelas positif. Selanjutnya label positif dan negatif dikonversi kedalam *binary*, nilai [1,0] untuk positif dan nilai [0,1] untuk negatif. Setelah dilakukan pemrosesan pada dataset, dataset dibagi kedalam 2 variabel berbeda. Variabel pertama adalah data latih dengan alokasi 90% dari jumlah data, sedangkan variabel kedua adalah data uji dengan alokasi 10%.

B. Implementasi perancangan sistem

1. Word2Vec

Penelitian ini menggunakan *pre-trained word2vec* yang didapat dari korpus wikipedia. Sebanyak 464.872 artikel diproses untuk membuat

3. Recall

Recall adalah sebuah matriks untuk model klasifikasi yang mengukur banyaknya prediksi positif yang didapat dari model yang sudah dibuat.

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

IV. HASIL DAN PEMBAHASAN

A. Dataset

Penelitian ini mengambil data dari twitter dengan kata kunci “anies baswedan” sebanyak 1019 data. Setelah data selesai diambil, data tersebut dilakukan pelabelan secara manual yang dilakukan oleh 3 orang mahasiswa termasuk penulis. Dari hasil pelabelan yang dilakukan, diperoleh data *tweet* sentimen positif dan negatif dengan distribusi sebagai berikut:

kamus kata yang akan digunakan untuk klasifikasi kata-kata. *Hyperparameter* yang digunakan pada pembangunan *word2vec* adalah sebagai berikut:

TABEL 7
HYPERPARAMETER WORD2VEC

No	Hyperparameter	Nilai
1	Jenis arsitektur <i>word2vec</i>	<i>Continuous bag-of-word (CBOW)</i>
2	Dimensi	100

2. RNN

Model sistem klasifikasi pada penelitian ini mempunyai spesifikasi sebagai berikut:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 30)]	0
embedding (Embedding)	(None, 30, 100)	310400
lstm (LSTM)	(None, 16)	7488
dense (Dense)	(None, 4)	68
dense_1 (Dense)	(None, 2)	10
Total params: 317,966		
Trainable params: 317,966		
Non-trainable params: 0		

GAMBAR 5
SPESIFIKASI MODEL

Pada penelitian ini, terdapat 5 layer pada model klasifikasi RNN. Layer pertama yaitu *input layer* dengan data yang diterima sebanyak 30 *sequence*. Layer kedua yaitu *embedding layer*. Fungsi dari *embedding layer* ini adalah untuk mencari kata-kata yang mempunyai konteks terdekat dalam ruang *embedding*. Layer ketiga yaitu *LSTM layer*. Layer keempat yaitu *dense layer*. *Dense layer* ini adalah layer yang fungsinya untuk menghubungkan setiap *neuron* dari layer sebelumnya ke layer setelahnya. Layer kelima yaitu *output layer*. *Output layer* fungsinya untuk menghasilkan *output* untuk klasifikasi.

Adapun hyperparameter yang digunakan pada sistem klasifikasi ini yaitu :

TABEL 8
HYPERPARAMETER RNN

No	Hyperparameter	Nilai
1	Dropout	0.5
2	Aktivasi <i>dense layer</i>	Rectified Linear Unit
3	Aktivasi <i>output layer</i>	Sigmoid
4	<i>Loss function</i>	<i>Binary crossentropy</i>
5	<i>Optimizer</i>	<i>Adam</i>
6	<i>Learning rate</i>	0.0007
7	<i>Epoch</i>	100
8	<i>Batch size</i>	32

C. Hasil implementasi model

Dalam penelitian ini, peneliti melakukan eksekusi terhadap model klasifikasi untuk mendapatkan performa yang dihasilkan oleh model klasifikasi. Berikut ini adalah hasil dari eksekusi terhadap model setiap kalinya:

TABEL 9
HASIL EKSEKUSI MODEL

No	Akurasi	Presisi	Recall	Confusion Matrix
1	0.8529	0.9130	0.7924	$\begin{bmatrix} 42 & 11 \\ 4 & 45 \end{bmatrix}$
2	0.8333	0.875	0.7924	$\begin{bmatrix} 42 & 11 \\ 6 & 43 \end{bmatrix}$
3	0.8529	0.88	0.8302	$\begin{bmatrix} 44 & 9 \\ 6 & 43 \end{bmatrix}$
4	0.8529	0.8654	0.8490	$\begin{bmatrix} 45 & 8 \\ 7 & 42 \end{bmatrix}$
5	0.8725	0.8846	0.8679	$\begin{bmatrix} 46 & 7 \\ 6 & 43 \end{bmatrix}$
6	0.8824	0.9020	0.8679	$\begin{bmatrix} 46 & 7 \\ 5 & 44 \end{bmatrix}$
7	0.8725	0.9167	0.8302	$\begin{bmatrix} 44 & 9 \\ 4 & 45 \end{bmatrix}$
8	0.8627	0.8980	0.8302	$\begin{bmatrix} 44 & 9 \\ 5 & 44 \end{bmatrix}$
9	0.8627	0.8846	0.8846	$\begin{bmatrix} 46 & 7 \\ 7 & 42 \end{bmatrix}$
10	0.8725	0.8704	0.8868	$\begin{bmatrix} 47 & 6 \\ 7 & 42 \end{bmatrix}$

D. Evaluasi model

Peneliti melakukan eksekusi terhadap model sistem klasifikasi sebanyak 10 kali. Performa yang diambil adalah performa rata-rata dari seluruh nilai yang diambil setiap eksekusi. Adapun rumus rata-rata adalah sebagai berikut:

$$A = \frac{1}{n} \sum_{i=1}^n a_i \quad (9)$$

Berdasarkan rumus rata-rata diatas, adapun hasil dari setiap *hyperparameter* adalah sebagai berikut:

1. Confusion matrix

TABEL 10
HASIL CONFUSION MATRIX

Prediksi			
Aktual	Positif	Positif	Negatif
	Negatif	45	8
		6	43

2. Akurasi

$$Akurasi = \frac{45+43}{45+6+43+8} = 0.8627 \quad (10)$$

3. Presisi

$$Presisi = \frac{45}{45+6} = 0.8824 \quad (11)$$

4. Recall

$$Recall = \frac{45}{45+8} = 0.8491 \quad (12)$$

V. KESIMPULAN

Berdasarkan hasil implementasi pada sistem yang sudah dibuat bahwa sistem yang telah dirancang yaitu analisis sentimen menggunakan metode *recurrent neural network* serta *word embedding word2vec* terkait isu Anies Baswedan sebagai calon presiden 2024 dapat dibangun dengan 1019 data yang bersumber dari sosial media *Twitter*. Pada penelitian ini dilakukan sebuah percobaan yaitu melakukan eksekusi sebanyak 10 kali untuk mengambil performa sistem setiap kali dilakukan eksekusi. Hasil yang didapat pada setiap kali eksekusi model sistem klasifikasi berbeda dikarenakan sistem klasifikasi RNN ini melakukan inisiasi terhadap bobot secara acak setiap kali dilakukan eksekusi. Penelitian ini menghasilkan nilai akurasi sebesar 86.27%, nilai *precision* sebesar 88.24%, dan nilai *recall* sebesar 84.91%. Hal ini membuktikan bahwa metode *recurrent neural network* yang dikombinasikan dengan *word embedding word2vec* dapat melakukan klasifikasi sentimen dengan baik.

Penulis menyarankan untuk penelitian selanjutnya agar dapat dilakukan penambahan data dan melakukan tuning hyperparameter supaya sistem menghasilkan performa lebih baik lagi.

REFERENSI

- [1] Y. Nugraha, 21 June 2022. [Online]. Available: <https://www.pikiran-rakyat.com/nasional/pr-014792940/hasil-survei-internal-pks-anies-baswedan-sosok-paling-banyak-dipilih-sebagai-calon-presiden>.
- [2] R. M. Wijaya, "MakassarTerkiniID," 1 November 2021. [Online]. Available: <https://makassar.terkini.id/hashtag-dukung-anies-presiden-2024-trending-twitter-netizen->

bandingkan-kepemimpinan-anies-dengan-ganjar-pranowo/. [Accessed 20 June 2022].

- [3] R. Nuryama, "TiNewss.com," 14 June 2022. [Online]. Available: <https://www.tinewss.com/indonesia-news/pr-1853618409/jumlah-pengguna-twitter-di-indonesia-pada-2022>. [Accessed 29 June 2022].
- [4] A. Patel and A. K. Tiwari, "Sentiment Analysis b using Recurrent Neural Network," 2019.
- [5] H. Yuan, Y. Wang, X. Feng and S. Su, "Sentiment Analysis Based on Weighted Word2vec and Att-LSTM," 2018.
- [6] S. S. Salim, A. N. Ghanshyam, D. M. Ashok, I. B. Mazahir and B. S. Thakare, "Deep LSTM-RN with Word Embedding for Sarcasm Detection o Twitter," 2020.
- [7] M. Z. Rahman, Y. A. Sari and N. Yudistira, "Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metod Long Short-Term Memory (LSTM)," 2021.
- [8] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan Metode LSTM Deep Learning," 2021.
- [9] M. R. F. Sya'bani, U. Enri and T. N. Padilal, "Analisis Sentimen Terhadap Bakal Calon Presiden 2024 dengan Algoritma Naive Bayes," 2022.
- [10] B. Liu, "Sentiment Analysis and Opinion Mining," 2012.
- [11] M. Kaur and M. Mohta, "A Review of Deep Learning with Recurrent Neural Network," 2019.
- [12] A. Graves, "Long Short Term Memory," in *Supervised Sequence Labelling with Recurrent Neural Network*, 2012, pp. 37-45.
- [13] P. Srivastava, "https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/", 10 December 2017. [Online]. [Accessed 1 September 2022].
- [14] X. Rong, "Word2Vec Parameter Learning Explained," 2014.
- [15] G. Futi, "https://www.researchgate.net/figure/Architecture-of-Word2Vec-models-CBOW-and-Skip-Gram_fig3_328373466," 1 January 2018. [Online]. [Accessed 19 September 2022].