

## SENTIMENT ANALYSIS OF INDONESIA'S CAPITAL RELOCATION USING WORD2VEC AND LONG SHORT-TERM MEMORY METHOD

Irma Yanti<sup>\*1</sup>, Ema Utami<sup>2</sup>

<sup>1,2</sup>Magister of Informatics Engineering Universitas Amikom Yogyakarta  
Email: <sup>1</sup>[irmayanti@students.amikom.ac.id](mailto:irmayanti@students.amikom.ac.id), <sup>2</sup>[ema.u@amikom.ac.id](mailto:ema.u@amikom.ac.id)

(Article received: August 26, 2024; Revision: September 20, 2024; published: February 20, 2025)

### Abstract

The relocation of the national capital (IKN) has garnered public attention, triggering various reactions and sentiments among the community. Sentiment analysis is crucial for understanding public perceptions of an issue, particularly on social media platforms like Twitter and YouTube. This study's sentiment analysis employs Word2Vec parameters, including architecture and dimensions. Additionally, hyperparameters such as the Optimizer and activation functions are applied to the Long Short-Term Memory (LSTM) model to analyze their effect on sentiment classification performance related to the IKN relocation. The study aims to compare the influence of Word2Vec parameters on LSTM model hyperparameter performance in sentiment classification. Data on the IKN relocation were gathered from tweets and YouTube video comments, then processed to form a text corpus used to train the Word2Vec model with Skip-gram and Continuous Bag-of-Words (CBOW) architectures, utilizing different dimension sizes (100 and 300) to enhance word representation in vectors. After obtaining word representations, the LSTM model was applied to classify sentiments using hyperparameters such as activation functions (ReLU, Sigmoid, and Tanh) and two Optimizers (Adam and RMSProp). The results indicate that the Skip-gram architecture tends to yield higher accuracy compared to CBOW, particularly with larger vector dimensions (300), which generally improved model accuracy, especially when using the RMSProp Optimizer and ReLU activation function, achieving an accuracy of 91%. It can be concluded that dimension values and architecture in Word2Vec, as well as the use of Optimizer and activation functions in LSTM, significantly impact model performance.

**Keywords:** Deep Learning, LSTM, Moving Capital, Text Classification, Word2vec

## ANALISIS SENTIMEN PEMINDAHAN IBU KOTA INDONESIA MENGGUNAKAN WORD EMBEDDING WORD2VEC DAN METODE LONG SHORT-TERM MEMORY

### Abstrak

Pemindahan Ibu Kota Negara (IKN) telah menjadi sorotan publik yang memicu berbagai reaksi dan sentimen di kalangan masyarakat. Analisis sentimen merupakan hal yang penting untuk memahami persepsi publik terhadap suatu isu, terutama di situs media sosial seperti Twitter dan YouTube. Analisis sentimen dalam penelitian ini menggunakan parameter *Word2Vec* yang mencakup arsitektur dan dimensi. Selain itu, penelitian ini juga menerapkan *hyperparameter* pada model *Long Short-Term Memory* (LSTM), seperti *Optimizer* dan fungsi aktivasi, untuk menganalisis pengaruhnya terhadap performa klasifikasi sentimen terkait pemindahan IKN. Tujuan dari penelitian ini untuk membandingkan pengaruh parameter *Word2Vec* terhadap performa pada *hyperparameter* model LSTM dalam klasifikasi sentimen. Data terkait pemindahan IKN dikumpulkan dari berbagai tweet dan komentar dari video YouTube, kemudian diproses untuk membentuk korpus teks yang digunakan untuk pelatihan model *Word2Vec* dengan arsitektur *Skip-gram* dan *Continuous Bag-of-Words* (CBOW), serta penggunaan ukuran dimensi yang berbeda (100 dan 300) untuk meningkatkan representasi kata dalam *vector*. Setelah representasi kata diperoleh, model LSTM diterapkan untuk mengklasifikasikan sentimen dengan menggunakan *hyperparameter* model LSTM meliputi; fungsi aktivasi (*ReLU*, *Sigmoid*, dan *Tanh*) serta dua *Optimizer* (*Adam* dan *RMSprop*). Dari hasil penelitian ini, arsitektur *Skip-gram* cenderung menghasilkan akurasi yang lebih tinggi dibandingkan CBOW, khususnya pada dimensi *vektor* yang lebih besar (300) hasilnya cenderung meningkatkan akurasi model, terutama saat menggunakan *Optimizer RMSProp* dan fungsi aktivasi *ReLU* didapatkan akurasi sebesar 91%. Dapat disimpulkan bahwa nilai dimensi dan arsitektur pada model *Word2Vec*, serta penggunaan *Optimizer* dan fungsi aktivasi pada LSTM dapat mempengaruhi kinerja model.

**Kata kunci:** *Deep Learning, LSTM, Moving Capital, Text Classification, Word2vec*

---

## 1. PENDAHULUAN

Presiden Joko Widodo memutuskan untuk memindahkan ibu kota negara Indonesia (IKN) ke luar Pulau Jawa dalam rapat pemerintah terbatas pada tanggal 29 April 2019. Keputusan ini tertuang dalam Rencana Pembangunan Jangka Menengah Nasional 2020–2024 [1]. Selama pemerintahan Presiden Joko Widodo, pemindahan IKN baru dikerjakan. Ini terbukti dengan pengesahan RUU IKN menjadi UU IKN pada tanggal 18 Januari 2022 [2]. Pemindahan IKN beralaskan pada visi misi Indonesia tahun 2045 yaitu Indonesia maju, ekonomi Indonesia akan masuk 5 besar dunia 2045 [3]. Pemindahan IKN memiliki potensi untuk meningkatkan pertumbuhan ekonomi nasional dengan tingkat inflasi yang tetap rendah. Selain itu, pembangunan kawasan industri dengan rantai nilai hingga hilir akan meningkatkan pendapatan masyarakat, yang pada gilirannya akan mengurangi perbedaan ekonomi antara Pulau Jawa dan luar Pulau Jawa [4]. Mengingat aspek historis, Palangkaraya dianggap sebagai pilihan yang menarik karena Presiden pertama Indonesia, Ir. Soekarno, karena dianggap cukup besar luas lahan untuk membangun infrastruktur yang diperlukan untuk keberadaan sebuah ibu kota negara [5]. Meskipun demikian, masalah pemindahan IKN ini jelas merupakan masalah yang sulit dan sangat penting bagi Indonesia, terutama dengan mempertimbangkan persaingan global [6]. Setiap orang memiliki pendapat yang berbeda tentang keputusan presiden Joko Widodo untuk memindahkan IKN, ada banyak orang yang merasa pro dan kontra dengan keputusan tersebut. Oleh karena itu perlu dilakukannya analisis sentimen terhadap pemindahan IKN untuk dapat diketahui permasalahan yang ada pada kontroversi pemindahan IKN, sehingga dapat dijadikan sebagai bahan evaluasi untuk kepentingan lebih lanjut

Analisis sentiment itu sendiri dapat digunakan untuk mengetahui kecenderungan pengguna Twitter ataupun Media Sosial lainnya untuk mengungkapkan pendapatnya [7]. Penelitian terkait sentimen analisis pemindahan IKN sebelumnya telah dilakukan dengan metode *machine learning*, seperti penelitian yang dilakukan oleh [8] sentimen publik yang bersumber dari data Twitter dengan menggunakan 2891 data. Penelitian ini membandingkan 4 algoritma konvensional meliputi pengklasifikasi Naïve Bayes, regresi logistik, support vector machine (SVM), dan K-nearest neighbor dengan menggunakan ekstraksi fitur TF-IDF. Perbandingan kinerja akurasi menunjukkan bahwa algoritma SVM menghasilkan nilai akurasi paling tinggi yaitu 97,72%. Penelitian terhadap pemindahan IKN juga pernah dilakukan oleh [9] dengan menggunakan

5046 dataset, namun menggunakan ekstraksi fitur *unigram* dengan menggunakan algoritma SVM, pada penelitian ini diperoleh akurasi 71,42%. Seiring dengan bertambahnya jumlah data, metode *machine learning* dianggap belum cukup dalam melakukan analisis sentimen [10]. Sehingga mulai banyak dilakukan penelitian dengan menggunakan metode *deep learning* karena cocok untuk penggunaan dataset dalam jumlah yang besar dan kompleks [11]. *Long Short-Term Memory* (LSTM) merupakan arsitektur RNN dengan memory cell yang lebih baik dan juga salah satu metode *deep learning* yang cukup sering digunakan untuk melakukan analisis sentiment [12]. Selain itu, keberhasilan *deep learning* tentang klasifikasi teks sangat bergantung pada efektivitas *word embedding* [13].

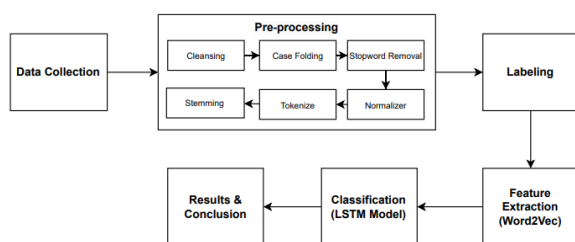
Beberapa peneliti telah menggunakan LSTM dengan fitur ekstraksi *word embedding* untuk analisis suatu text dengan tujuan memahami bagaimana respon seseorang terhadap suatu topik, produk, atau layanan. Pada penelitian sebelumnya yang melibatkan penambahan kata-kata, para peneliti melakukan penelitian terhadap penggunaan model LSTM dan parameter *Word2Vec* yang terdiri dari arsitektur yaitu *Skip-gram* dan *Continuous Bag-of-Words* (CBOW) untuk menentukan persentase ulasan yang berdampak pada layanan hotel di Indonesia [14]. Parameter model *Word2Vec* terdiri dari arsitektur (*Skip-gram* dan CBOW), teknik evaluasi (*Negative Sampling* dan *Hierarchical Softmax*), dan dimensi vektor (100, 200, dan 300). Parameter LSTM terdiri dari nilai *dropout* yang digunakan yaitu 0,2, 0,5, dan 0,7, nilai learning rate 0.0001 dan 0.001, *pooling method* (maksimum dan rata-rata pooling). skenario terbaik dihasilkan dengan menggunakan *Skip-Gram*, *Hierarchical Softmax* sebagai metode evaluasi, dan dengan nilai dimensi vektor yang diatur ke 300. Selain itu juga penggunaan model LSTM and *Word2Vec* dilakukan oleh [15]. Penelitian yang diusulkan dievaluasi dengan menggunakan dua teknik penyisipan kata: *Word2Vec* dengan *Skip-gram* dan *Word2Vec* dengan CBOW. Berdasarkan hasil penelitian ini, LSTM dengan *Word2Vec* CBOW mencapai hasil yang lebih baik dibandingkan dengan LSTM menggunakan *Word2Vec Skip-gram*. Penelitian yang dilakukan [16] juga menunjukkan bahwa metode CBOW memberikan hasil yang lebih baik ketika menggunakan kombinasi parameter evaluasi *Hierarchical Softmax* dan dimensi vektor sebesar 100. Selanjutnya [17] melakukan penelitian pada klasifikasi text dataset *Turkish Text Classification* 3600 (TTC-3600) yang terdiri dari teks berita Turki dan dataset *BBC-News* berita bahasa Inggris, penelitian ini mengusulkan model *deep learning* yaitu *Convolutional Neural Network* (CNN) dan

juga penambahan model ekstraksi fitur dengan menggunakan *Doc2Vec*. tingkat akurasi, 94,17% diperoleh sebagai hasil dari pengklasifikasian dataset Clean + Stem-DS yang dibuat dari dataset TTC-3600 dengan CNN dan pengklasifikasian dataset *BBC-News*, diperoleh akurasi 96,41%. Hasil penelitian lainnya Model *Word2Vec* yang memberikan akurasi tertinggi adalah model dengan arsitektur *Skip-gram*, yang dievaluasi menggunakan metode *Hierarchical Softmax*, dan memiliki dimensi vektor sebesar 100, dimensi vektor juga mempengaruhi akurasi model. Untuk mendapatkan hasil yang optimal, perlu meningkatkan dimensi vektor dan jumlah data yang digunakan dalam pelatihan secara bersamaan [18].

Untuk menganalisis dampaknya terhadap performa klasifikasi sentimen terkait pemindahan IKN, penelitian ini menggunakan parameter *Word2Vec* yang meliputi arsitektur dan dimensi. Selain itu juga menerapkan *hyperparameter* pada model LSTM, seperti *optimizer* dan fungsi aktivasi. Tujuan penelitian ini yaitu untuk membandingkan pengaruh parameter *Word2Vec* terhadap performa pada *hyperparameter* model LSTM dalam klasifikasi sentimen

## 2. METODE PENELITIAN

Bagian ini menjelaskan tahapan yang dilakukan dalam penelitian, antara lain; *Data Collection*, *Preprocessing*, *Labeling*, *Feature Extraction*, *Classification model* dan *Evaluasi Hasil*. Gambar 1 menunjukkan alur yang dilakukan dalam penelitian. Selanjutnya untuk detail tahapan penelitian serta metode yang digunakan akan dibahas pada masing-masing sub bab.



Gambar 1. Alur Penelitian

### 2.1. Data Collection

Proses pengumpulan data dilakukan dengan menggunakan library *Tweet Harvest* dan API YouTube menggunakan kata kunci yang berkaitan dengan pemindahan IKN, seperti: pemindahan ibu kota, relokasi IKN, pembangunan IKN, dan relokasi Jakarta ke IKN. Data yang dikumpulkan mencakup tweet dan komentar video. Hasil dari proses pengambilan data sebanyak 9.842 data, dengan rincian 4.300 data diperoleh dari Twitter dan 5.542 data diperoleh dari YouTube. Semua data tersebut disimpan dalam format CSV. Gambar 2 merupakan contoh dataset yang berhasil dikumpulkan.

@Miduk17 Cuma Anis yang TOLak jagoanmu si Uban koyok yes yes o lanjut IKN  
 @tanyarifes Tolak masal aja mereka butuh buat bangun IKN  
 mungkin kita bisa selamat kalo tolak IKN tolak program pemerintah ga jelas menghambur2kan apbn kaya program makan gratis untuk pembangunan ugala2 lainnya. Kalo ada demo gw dukung dah 1000%  
 @jokowi Tolak IKN !!  
 @Kimberley\_PS08 Bukan hanya tolak tapi lawan untuk proyek sampul paspor baru tapi smua ikn tapera kris

Gambar 2. Hasil *Data Collection*

### 2.2. Preprocessing

*Preprocessing* merupakan tahapan untuk mengubah data tweet yang tidak terstruktur menjadi data yang baik dan siap untuk diolah dengan banyak *cluster* [19]. Berikut ini adalah langkah-langkah yang diterapkan dalam pemrosesan data text [20]:

#### a. Case folding

*Case folding* merupakan proses mengubah semua huruf dalam teks menjadi format huruf kecil (*lowercase*), tujuannya untuk menyamakan format teks dan mengurangi variasi data yang tidak diperlukan.

#### b. Cleansing

*Cleansing* merupakan proses pembersihan data seperti menghapus *username*, *hashtag*, URL, *retweet*, dan *punctuation*.

#### c. Stopword Removal

*Stopword Removal* adalah proses menghilangkan kata-kata yang dianggap tidak memiliki makna penting dengan menggunakan Pustaka Sastrawi. proses ini dilakukan agar ukuran data yang diproses menjadi lebih kecil/sedikit, sehingga algoritma dapat fokus pada kata-kata yang lebih relevan untuk analisis.

#### d. Normalization

*Normalization* dilakukan untuk menstandarkan bentuk kata yang awalnya tidak baku (*slang word*) menjadi kata baku. Selain itu juga dapat memperbaiki kata-kata singkatan.

#### e. Tokenize

*Tokenize* adalah proses untuk memecah sebuah kalimat kedalam bentuk kata per kata sehingga memudahkan untuk diproses lebih lanjut.

#### f. Stemming

Proses *Stemming* mengubah kata yang berimbuhan menjadi kata dasar.

### 2.3. Labeling

Setelah data selesai di *Processing* selanjutnya diberikan label untuk mempresiksi data termasuk kedalam komentar positif, negatif atau netral. Penelitian ini menggunakan *Lexicon Based* untuk melakukan pelabelan data secara otomatis. Kamus *Lexicon* berisikan bobot dari sebuah kata yang digunakan sebagai acuan untuk menghitung bobot dari sebuah kata perhitungan [21].

### 2.4. Feature Extraction (*Word2Vec*)

Setelah data diberikan label selanjutnya data yang berisi komentar direpresentasikan menjadi

sebuah kata-kata ke dalam bentuk *vector* menggunakan ekstraksi fitur model *Word2Vec*. *Word2Vec* diperkenalkan oleh Mikolov pada tahun 2013, di mana korpus digunakan sebagai input dan menghasilkan output berupa vektor. *Word2Vec* mengubah setiap kata unik menjadi vektor. Salah satu keunggulan *Word2Vec* adalah kemampuannya merepresentasikan kesamaan kontekstual antara dua kata dalam vektor yang dihasilkan [22].

*Word2Vec* adalah algoritma yang menghasilkan representasi vektor untuk kata-kata dan dapat mencapai kinerja optimal dalam *Natural Language Processing* (NLP) dengan mengelompokkan kata-kata yang mirip ke dalam vektor yang sama. *Word2Vec* menggunakan *neural network* untuk menghitung representasi kata dalam bentuk vektor. Vektor yang dihasilkan berada dalam ruang dimensi yang mampu menangkap makna semantik dari kata tersebut [23]. *Word2Vec* memiliki sejumlah parameter yang mempengaruhi proses pembelajaran model, seperti arsitektur dan dimensi. Setiap jenis parameter tersebut berdampak pada tingkat akurasi dalam performa model *deep learning*. *Word2Vec* dapat dihitung dengan menggunakan nilai vektor kata yang diperoleh dengan menghitung *Cosine Similarity* antara dua vektor n-dimensi dengan mencari nilai cosinus dari sudut antara keduanya [24] dengan menggunakan persamaan berikut:

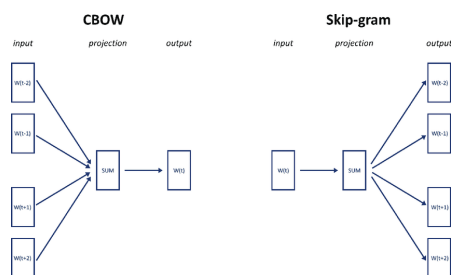
$$Similarity = \cos \theta = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \quad (1)$$

Hasil dari Persamaan rumus diatas kemudian dihitung ulang menggunakan Persamaan Korelasi Pearson untuk mengevaluasi hasil perhitungan kemiripan dengan menggunakan persamaan berikut:

$$corr = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (2)$$

a. Arsitektur *Word2Vec*

*Word2Vec* memiliki dua arsitektur yaitu *Continuous Bag of Words* (CBOW) dan *Skip-gram* [25]. Pada arsitektur *Skip-gram* memprediksi konteks kata-kata di sekitar kata target tertentu, sedangkan pada arsitektur CBOW model memprediksi kata target berdasarkan konteks kata di sekitarnya [26]



Gambar 3. Arsitektur *Word2Vec*

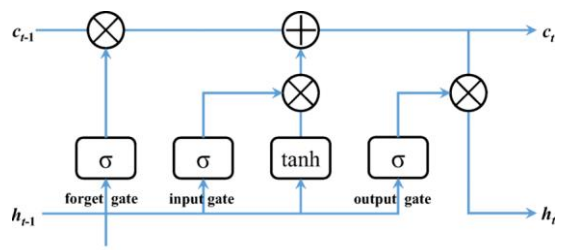
b. Dimensi *Word2Vec*

Dimensi *Word2Vec* sangat dipengaruhi oleh korpus yang digunakan saat pelatihan. Dimensi yang terlalu kecil tidak akan mampu menangkap makna semantik dari korpus, sementara dimensi yang terlalu besar dapat memperlambat proses pemrosesan. Oleh karena itu, menentukan dimensi yang tepat sangat penting dalam mengubah kata menjadi vector [27].

Dalam pembuatan model *Word2Vec*, korpus merupakan hal sangat penting karena keragaman kata dalam korpus akan mempengaruhi kualitas model yang dihasilkan [28]. Pembentukan korpus dalam penelitian ini menggunakan *liblary Gensim*. Model *Word2Vec* selanjutnya diatur dengan menentukan dimensi vektor dan ukuran jendela. Pada penelitian ini, dimensi vektor yang digunakan adalah 100 dan 300, untuk ukuran jendela 10.

2.5. *Long Short-Term Memory (LSTM)*

Tujuan LSTM adalah untuk memecahkan masalah RNN. *Gradient vanishing* dan *exploding LSTM* menggantikan vektor tersembunyi RNN dengan blok memori yang dilengkapi gerbang. Pada dasarnya, dengan melatih bobot rating yang tepat, dapat mempertahankan memori jangka panjang dan telah terbukti sangat bermanfaat untuk memecahkan masalah kontemporer, seperti pengenalan ucapan [29]. LSTM memiliki beberapa lapisan yaitu *Input gate*, *Forget gate*, *Output gate* dan *Hidden layer* seperti pada Gambar 4. *Input gate* berfungsi untuk menentukan nilai input yang akan diperbarui pada *state memory*. *Forget gate* berfungsi untuk menentukan informasi yang dihapus dari *cell*. *Output gate* berfungsi untuk menentukan output yang dihasilkan apakah sesuai dengan input dan memori pada *cell*. *Hidden layer* berfungsi sebagai lapisan pemrosesan utama yang menyimpan informasi dari urutan data sebelumnya [30].



Gambar 4. Arsitektur LSTM

Berikut adalah persamaan dari setiap lapisan LSTM:

1. Input Gate

$$i_t = \sigma(x_t \cdot w_{ix} + h_{t-1} \cdot w_{ih} + b_i) \quad (3)$$

2. Forget Gate

$$f_t = \sigma(x_t \cdot w_{fx} + h_{t-1} \cdot w_{fh} + b_f) \quad (4)$$

3. Output Gate

$$o_t = \sigma(x_t \cdot w_{ox} + h_{t-1} \cdot w_{oh} + b_o) \quad (5)$$

4. Hiden Layer  

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

Langkah pertama pembuatan model LSTM dalam penelitian ini yaitu menerapkan *layer embedding* untuk mengubah kata-kata menjadi vektor berdasarkan model *Word2Vec* yang sudah dilatih sebelumnya. Setelah itu, ditentukan optimizer dan nilai learning rate. Model LSTM kemudian dilatih menggunakan data pelatihan secara berulang (*epoch*). Selama proses pelatihan, model juga diuji untuk melihat seberapa baik kinerja pada data yang belum pernah dilihat sebelumnya, dan sistem mencatat hasil performa model setiap kali selesai satu *epoch*. Tabel 1 mencantumkan *hyperparameter* LSTM yang digunakan dalam penelitian ini.

Tabel 1. *Hyperparameter* LSTM

No	Parameter	Value
1	Oprimizer	Adam, RMSProp
2	Hiden Layer	Relu, Sigmoid, Tahn
3	Output Layer	Softmax
4	Learning Rate	0,01
5	Batch Size	64
6	Epoch	15
7	Hiden unit	100

**2.6. Metode Pengujian**

Tabel 2 menunjukkan skema pengujian Confusion Matrix yang digunakan dalam penelitian ini. Metode pengujian kinerja ini merupakan tabel untuk mengukur kinerja algoritma klasifikasi atau model klasifikasi atau classifier [25].

Tabel 2. Confussion Matrix

	Actual (+)	Actual (-)
Prediksi Class (+)	True Positive(TP)	False Positive(FP)
Prediksi Class (-)	False Negative(FN)	True Negative(TN)

Pengukuran *confusion matrix* menggunakan nilai akurasi, yang dihitung dengan menggunakan persamaan berikut:

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \tag{7}$$

**3. HASIL DAN PEMBAHASAN**

Hasil penelitian akan dibahas di bagian ini mulai dari proses pengumpulan data, hasil *Preprocessing*, hasil kinerja terhadap penggunaan *Word2Vec* dan metode LSTM dalam menganalisis dataset yang bersumber dari twitter dan youtube yang berhubungan dengan pemindahan IKN, dan juga menjelaskan perbandingan akurasi terhadap penggunaan arsitektur *Wod2vec* dan fungsi aktivasi pada LSTM.

**3.1. Data Collection**

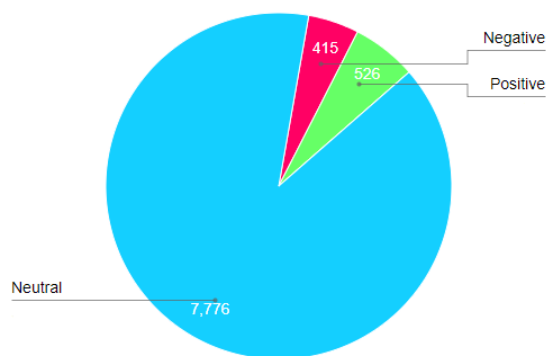
Data dikumpulkan berasal dari *Twitter* dan komentar dari video Youtube yang membahas isu pemindahan Ibu Kota Negara. Data yang berhasil

dikumpulkan berjumlah 9.842, namun sebanyak 1.125 terindikasi sebagai data duplikat, sehingga dataset yang akan dilakukan *Preprocessing* sebanyak 8.717 data. Hasil *Preprocessing* ditunjukkan pada tabel 3.

Tabel 3. Hasil *Preprocessing*

Process	Sentences
Clawing	@Kimberley_PS08 Bukan hanya tolak tapi lawan untuk proyek sampul paspor baru tapi smua ikn tapera kris.
Case Folding	@Kimberley_PS08 bukan hanya tolak tapi lawan untuk proyek sampul paspor baru tapi smua ikn tapera kris
Cleansing	bukan hanya tolak tapi lawan untuk proyek sampul paspor baru tapi smua ikn tapera kris
Stopword Removal	tolak lawan proyek sampul paspor smua ikn tapera kris
Normalization	tolak lawan proyek sampul paspor smua ikn tapera kris
Tokenize	['tolak', 'lawan', 'proyek', 'sampul', 'paspor', 'semua', 'ikn', 'tapera', 'kris']
Stemming	['tolak', 'lawan', 'proyek', 'sampul', 'paspor', 'semua', 'ikn', 'tapera', 'kris']

Selanjutnya data diberi label dengan menggunakan *library VaderSentiment* untuk mengelompokan data menjadi tiga kategori yaitu data positif, negatif dan netral berdasarkan Gambar 5.



Gambar 5. Hasil *Labeling Data*

**3.2. Feature Extraction (Word2Vec)**

Penelitian ini menggunakan arsitektur *Skip-gram* dan *CBOw* dengan ukuran dimensi 100 dan 300. Tabel 4 menunjukkan kata yang mempunyai kemiripan dengan kata “Jakarta” dan nilai *similarity* pada masing-masing kata.

Tabel 4. Perbandingan Dimensi Arsitektur *CBOw* & *Skip-gram*

Architecture	Dimensions	Most Similar	Similarity Score
Skip-gram	100	dki	0.7164506316184
		jkt	0.6194050908088
		pikuk	0.6293399333953
CBOw	100	hiruk	0.6042805314064
		dki	0.9395583868026
		pemprov	0.8662981986999
		macet	0.8352913260459
		padat	0.8569363355636

Skip-gram		dki	0.7544746398925
		jkt	0.6373074054718
CBOW	300	pikuk	0.6563802957534
		hiruk	0.6431236267089
		dki	0.9470203518867
		pemprov	0.8889915943145
		klasik	0.8597777485847
		padat	0.8439351320266

Pada arsitektur *Skip-gram* dengan dimensi 100 kata-kata yang paling mirip adalah “dki”, “jkt”, “pikuk”, “hiruk”, dan “macet” dengan *similarity score* tertinggi adalah 0.716. Pada arsitektur CBOW kata-kata yang paling mirip juga mencakup “dki”, “jkt”, “hiruk”, “pemprov”, dan “padat” dengan *similarity score* tertinggi adalah 0.939, dari hasil tersebut menunjukkan bahwa arsitektur CBOW lebih baik dalam menangkap kesamaan kata pada dimensi ini. Sedangkan pada dimensi 300 arsitektur dengan arsitektur *Skip-gram* kata-kata yang paling mirip adalah “dki”, “jkt”, “pikuk”, “hiruk”, dan “macet” dengan *similarity score* tertinggi adalah 0.754. Pada arsitektur CBOW kata-kata yang paling mirip mencakup “dki”, “pemprov”, “klasik”, dan “padat”, *similarity score* tertinggi adalah 0.947, skor pada dimensi 300 sedikit lebih tinggi dibandingkan dengan model dimensi 100. Sehingga dapat disimpulkan bahwa arsitektur CBOW cenderung memberikan *similarity score* yang lebih tinggi dibandingkan dengan *Skip-gram* baik pada dimensi 100 maupun 300 dan meningkatkan dimensi *vector* dari 100 ke 300, cenderung meningkatkan *similarity score* baik pada *Skip-gram* ataupun CBOW.

### 3.3. Hasil Evaluasi Word2Vec Skip-gram dengan Hyperparameter LSTM

Model *Word2vec* dengan arsitektur *Skip-Gram* menggunakan nilai dimensi 100 dan 300, kemudian mengukur performa model LSTM dengan melakukan perbandingan pada fungsi aktivasi dan *Optimizer*. Model diuji dengan data yang telah dibagi dengan perbandingan 20% untuk data uji dan 80% untuk data latih.

Tabel 5. Hasil Evaluasi *Skip-gram* & *Hyperparameter LSTM*

Optimizer	Activation	Dimensions	Accuracy
RmsProp	Relu	100	89
	Sigmoid		88
	Tahn		89
Adam	Relu	100	88
	Sigmoid		88
	Tahn		89
<b>RmsProp</b>	<b>Relu</b>	300	<b>91</b>
	Sigmoid		90
	Tahn		90
Adam	Relu	300	90
	Sigmoid		90
	Tahn		90

Evaluasi kinerja pada arsitektur *Skip-gram* pada dimensi 300 dengan menggunakan *Optimizer RMSProp* dan fungsi aktivasi *ReLU* menghasilkan akurasi tertinggi yaitu sebesar 91% yang ditunjukkan pada Tabel 5. Ini menunjukkan keunggulan kombinasi tersebut dalam menangkap makna kata pada dimensi yang lebih besar. Berbeda dengan

*Optimizer Adam*, disemua fungsi aktivasi menunjukkan akurasi yang sama yaitu 90%. Ini membuktikan bahwa *Optimizer Adam* memberikan kinerja yang konsisten di berbagai fungsi aktivasi pada dimensi 300. Pada dimensi 100 dengan menggunakan *Optimizer RMSProp*, fungsi aktivasi *ReLU* dan *Tanh* menghasilkan akurasi tertinggi sebesar 89%. Ini menandakan bahwa kedua fungsi aktivasi ini cukup efektif pada dimensi yang lebih kecil. Sedangkan untuk *Optimizer Adam*, akurasi tertinggi sebesar 89% diperoleh dengan menggunakan fungsi aktivasi *Tanh*. Ini menunjukkan bahwa fungsi aktivasi *Tanh* sedikit lebih unggul dibandingkan *ReLU* dan *Sigmoid* saat menggunakan *Optimizer Adam* pada dimensi 100.

### 3.4. Hasil Evaluasi Word2Vec CBOW dengan Hyperparameter LSTM

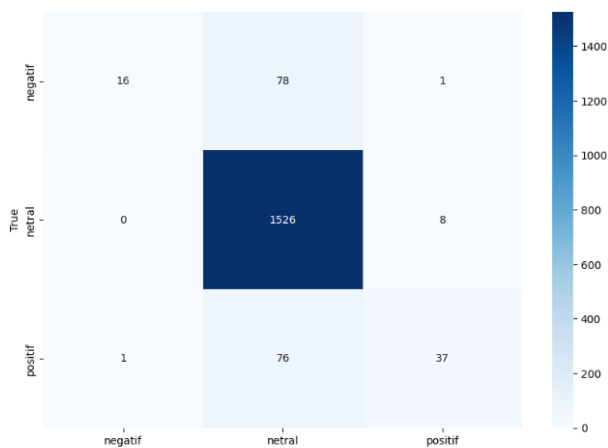
Berbeda dengan hasil evaluasi pada arsitektur CBOW yang dapat dilihat pada Tabel 6, dimana tidak ada perbedaan yang signifikan antara kinerja *Optimizer RMSProp* dan *Adam*. Peningkatan dimensi vektor yang lebih besar (300) tidak memberikan peningkatan akurasi yang signifikan dibandingkan dengan dimensi yang lebih kecil (100) pada arsitektur ini.

Tabel 6. Hasil Evaluasi CBOW & *Hyperparameter LSTM*

Optimizer	Activation	Dimension	Accuracy
RmsProp	Relu	100	89
	Sigmoid		88
	Tahn		88
Adam	Relu	100	89
	Sigmoid		88
	Tahn		89
RmsProp	Relu	300	89
	Sigmoid		88
	Tahn		88
Adam	Relu	300	88
	Sigmoid		89
	Tahn		89

### 3.5. Pengujian Confusion Matrix

Hasil pengujian pada kombinasi optimasi *RMSProp*, fungsi aktivasi *ReLU* dan arsitektur *Skip-gram* untuk model klasifikasinya ditunjukkan pada Gambar 6 melalui *confusion matrix*. Model memiliki performa yang cukup baik dalam mengklasifikasikan kelas netral dengan angka prediksi benar yang tinggi yaitu 1526. Namun, model memiliki kesulitan dalam mengklasifikasikan kelas negatif dan positif dengan benar, seperti terlihat dari banyaknya kesalahan prediksi pada kedua kelas tersebut. Kelas negatif sering salah diprediksi sebagai netral, dan kelas positif juga sering salah diprediksi sebagai netral.



Gambar 6. Pengujian *Confusion Matrix*

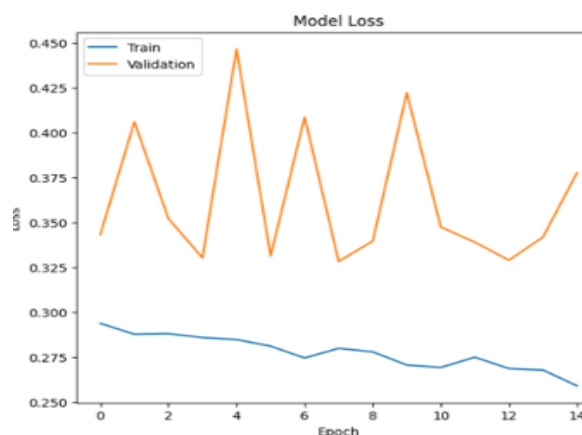
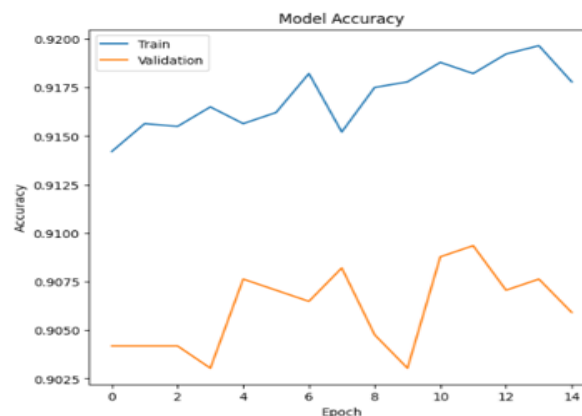
Berdasarkan confusion matrix tersebut, Gambar menunjukkan metrik evaluasi yang dihasilkan dengan mencakup berbagai aspek penting dalam penilaian performa model, seperti akurasi, presisi, recall, dan F1-score. Dapat dilihat pada Gambar 7, untuk kelas netral, model menunjukkan performa yang sangat baik dalam mengenali dan memprediksi data dengan label netral, dengan nilai precision 0,91, recall 0,99, dan F1-score 0,95. Selanjutnya untuk performa model kelas positif lebih rendah dibandingkan kelas netral dengan precision sebesar 0.80 dan recall sebesar 0.32, model cenderung mengalami kesulitan untuk mengenali data positif, yang menghasilkan F1-score yang rendah sebesar 0.46. Di antara semua kelas performa kelas negatif terendah, dengan recall hanya 0.17, hal ini menunjukkan bahwa model tidak dapat mendeteksi banyak data negatif. Meskipun kelas ini memiliki tingkat keakuratan yang cukup tinggi (0.94), nilai F1-score hanya 0.29 karena tingkat recall yang rendah.

	precision	recall	f1-score	support
negatif	0.94	0.17	0.29	95
netral	0.91	0.99	0.95	1534
positif	0.80	0.32	0.46	114
accuracy			0.91	1743
macro avg	0.88	0.50	0.57	1743
weighted avg	0.90	0.91	0.88	1743

Gambar 7. Performa Klasifikasi Model

### 3.6. Evaluasi Model Accuracy & Loss

Gambar 8 menunjukkan performa model selama proses pelatihan dan validasi terhadap *Epoch*. Pada arsitektur *Skip-gram* menunjukkan bahwa dimensi 300 dengan *Optimizer RMSProp* dan fungsi aktivasi *ReLU*, akurasi pada data training meningkat secara konsisten, tetapi akurasi pada data *Validation* tidak stabil dan cenderung lebih rendah, selain itu nilai *Loss* pada data *Validation* lebih tinggi dibandingkan dengan *Training Loss*.



Gambar 8. Model Accuracy and Loss pada *Skip-gram* dimensi 300

## 4. DISKUSI

Penelitian ini melakukan evaluasi kinerja model *Long Short-Term Memory* (LSTM) yang digabungkan dengan *Word2Vec* seperti yang pernah dilakukan oleh penelitian sebelumnya [14] [15] [17] [18] namun objek yang digunakan tidak sama. Peneliti melakukan analisis sentimen masyarakat terhadap pemindahan Ibu Kota Negara Indonesia dengan melakukan konfigurasi dimensi *vector*, fungsi aktivasi, serta optimisasi yang bertujuan untuk membandingkan. dalam penelitian ini menemukan bahwa arsitektur *Skip-gram* secara konsisten menghasilkan akurasi yang lebih tinggi dibandingkan CBOV, terutama ketika menggunakan dimensi vektor yang lebih besar. Ini sejalan dengan penelitian yang sudah dilakukan sebelumnya [14] yang juga menunjukkan bahwa penggunaan arsitektur *Skip-gram* dengan dimensi vektor lebih besar mampu menangkap representasi kata yang lebih baik dalam konteks yang lebih luas, tetapi jumlah data yang digunakan lebih sedikit dibandingkan dengan penelitian ini. Berbeda dengan penelitian [15] yang menunjukan bahwa arsitektur CBOV memberikan hasil yang lebih baik jika dibandingkan dengan *Skip-gram*, hal ini karena penelitian [15] menggunakan ukuran *max length* yang besar yaitu 10.000, sehingga CBOV lebih baik dalam menangkap konteks kata yang lebih luas.

Sedangkan dalam penelitian ini hanya menggunakan ukuran *max length* 100, sehingga hanya mampu menangkap konteks kata yang paling dekat dengan kata target.

Penelitian ini menghasilkan akurasi yang baik khususnya pada saat penggunaan kombinasi optimasi *RMSProp*, fungsi aktivasi *ReLU* dan arsitektur *Skip-gram*. Namun peneliti juga mengidentifikasi adanya *Overfitting*, yang diindikasikan oleh perbedaan performa antara data training dan data validasi yang dapat dilihat pada gambar 5. Hal ini menunjukkan bahwa, meskipun model yang digunakan berhasil menangkap pola pada data latih dengan baik, tetapi performa tidak selalu konsisten.

## 5. KESIMPULAN

Dari hasil penerapan model *Word2vec* dengan arsitektur *Skip-gram* dan CBOW dengan dimensi yang berbeda (100, 300) serta membandingkan beberapa fungsi aktivasi (*ReLU*, *Sigmoid*, *Tahn*) dan *Optimizer* (*RMSProp*, *Adam*) pada model LSTM dengan menggunakan dataset yang diambil dari Twitter dan Youtube terkait tanggapan masyarakat tentang pemindahan Ibu Kota Negara, dapat disimpulkan bahwa arsitektur *Skip-gram* cenderung menghasilkan akurasi yang lebih tinggi dibandingkan CBOW, terutama ketika menggunakan dimensi vektor yang lebih besar, artinya arsitektur *Skip-gram* dengan dimensi vektor yang lebih besar mampu menangkap representasi kata yang lebih baik dalam konteks yang lebih luas. Penelitian ini menghasilkan akurasi yang baik khususnya pada saat penggunaan kombinasi optimasi *RMSProp*, fungsi aktivasi *ReLU* dan arsitektur *Skip-gram* yaitu sebesar 91%. Penggunaan *Optimizer* dan fungsi aktivasi juga dapat mempengaruhi kinerja model. Namun peneliti mengidentifikasi adanya *Overfitting*, yang diindikasikan oleh perbedaan performa antara data training dan data validasi yang dapat dilihat pada Gambar 6. Kedepannya, penelitian ini akan melakukan perbandingan dengan seleksi fitur lainnya, untuk mengidentifikasi metode seleksi fitur yang paling efektif untuk meningkatkan kinerja model dan juga mempertimbangkan penggunaan teknik lainnya untuk mengurangi *Overfitting*.

## DAFTAR PUSTAKA

- [1] Yasonna H. Laoly, "Presiden Republik Indonesia Peraturan Presiden Republik Indonesia," 2020.
- [2] N. Suchyo *et al.*, "Analisis sentimen masyarakat terhadap uu cipta kerja pada media sosial twitter," 2022, doi: 10.56486/jris.vol2no1.167.
- [3] A. Rahman Hakim, W. Gata, A. Zevana Putri Widodo, O. Kurniawan, and A. Rama Syarif, "Analisis Perbandingan Algoritma Machine Learning Terhadap Sentimen Analisis Pemindahan Ibu Kota Negara," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 7, no. 2, 2023, doi: 10.35870/jti.
- [4] S. Aditua Fandhitya Silalahi, "Kajian singkat terhadap isu aktual dan strategis dampak ekonomi dan risiko pemindahan ibu kota negara," 2019. doi: <http://puslit.dpr.go.id>.
- [5] Nicodemus R Toun, "Analisis Kesiapan Pemerintah Provinsi Kalimantan Tengah dalam Wacana Pemindahan Ibu Kota Negara Republik Indonesia ke Kota Palangkaraya Nicodemus R Toun," 2018. doi: 10.36859/jap.v1i01.45.
- [6] N. S. Wardani, A. Prahutama, and P. Kartikasari, "Analisis Sentimen Pemindahan Ibu Kota Negara Dengan Klasifikasi Naive Bayes Untuk Model Bernoulli dan Multinomial," 2020, doi: 10.14710/j.gauss.v9i3.27963.
- [7] S. Fendyputra Pratama, R. Andrian, and A. Nugroho, "Analisis Sentimen Twitter Debat Calon Presiden Indonesia Menggunakan Metode Fined-Grained Sentiment Analysis," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 4, no. 2, pp. 2541–3619, 2019, doi: 10.31328/jo.
- [8] E. Sutoyo and A. Almaarif, "Twitter sentiment analysis of the relocation of Indonesia's capital city," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1620–1630, Aug. 2020, doi: 10.11591/eei.v9i4.2352.
- [9] A. Huwaidah, Adiwijaya, and S. Al Faraby, "Argument Identification in Indonesian Tweets on the Issue of Moving the Indonesian Capital," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 407–415. doi: 10.1016/j.procs.2021.01.023.
- [10] C. Liu, P. Zhang, T. Li, and Y. Yan, "Semantic features based N-Best rescoring methods for automatic speech recognition," *Applied Sciences (Switzerland)*, vol. 9, no. 23, Dec. 2019, doi: 10.3390/app9235053.
- [11] J. (Consultant) Patterson and A. Gibson, *Deep learning : a practitioner's approach*.
- [12] A. M. Bahador, "The accuracy of the LSTM model for predicting the S&P 500 index and the difference between prediction and backtesting," 2018.
- [13] D. Shen *et al.*, "Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms."
- [14] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews," in *Procedia Computer Science*, Elsevier B.V.,



- 2021, pp. 728–735. doi: 10.1016/j.procs.2021.01.061.
- [15] S. Amin, M. Irfan Uddin, M. Ali Zeb, A. A. Alarood, M. Mahmoud, and M. H. Alkinani, “Detecting dengue/flu infections based on tweets using LSTM and word embedding,” *IEEE Access*, vol. 8, pp. 189054–189068, 2020, doi: 10.1109/ACCESS.2020.3031174.
- [16] D. Intan Af *et al.*, “Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen,” vol. 6, no. 3, 2021.
- [17] H. B. Dogru, S. Tilki, A. Jamil, and A. Ali Hameed, “Deep Learning-Based Classification of News Texts Using Doc2Vec Model,” in *2021 1st International Conference on Artificial Intelligence and Data Analytics, CAIDA 2021*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 91–96. doi: 10.1109/CAIDA51941.2021.9425290.
- [18] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, “Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 360–366. doi: 10.1016/j.procs.2019.08.178.
- [19] M. I. Hutapea and A. P. Silalahi, “Moderna’s Vaccine Using the K-Nearest Neighbor (KNN) Method: An Analysis of Community Sentiment on Twitter,” *Jurnal Penelitian Pendidikan IPA*, vol. 9, no. 5, pp. 3808–3814, May 2023, doi: 10.29303/jppipa.v9i5.3203.
- [20] A. Kadhim, “An Evaluation of Preprocessing Techniques for Text Classification,” 2018. [Online]. Available: <https://sites.google.com/site/ijcsis/>
- [21] G. D’Aniello, M. Gaeta, and I. La Rocca, “KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis,” *Artif Intell Rev*, vol. 55, no. 7, pp. 5543–5574, Oct. 2022, doi: 10.1007/s10462-021-10134-9.
- [22] Y. Zhu, E. Yan, and F. Wang, “Semantic relatedness and similarity of biomedical terms: Examining the effects of recency, size, and section of biomedical publications on the performance of word2vec,” *BMC Med Inform Decis Mak*, vol. 17, no. 1, Jul. 2017, doi: 10.1186/s12911-017-0498-1.
- [23] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, “Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 360–366. doi: 10.1016/j.procs.2019.08.178.
- [24] S. Lai, K. Liu, L. Xu, and J. Zhao, “How to Generate a Good Word Embedding?,” Jul. 2015, [Online]. Available: <http://arxiv.org/abs/1507.05523>
- [25] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, “Word2vec model analysis for semantic similarities in English words,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 160–167. doi: 10.1016/j.procs.2019.08.153.
- [26] Y. Ardian Pradana, I. Cholissodin, and D. Kurnianingtyas, “Analisis Sentimen Pemindahan Ibu Kota Indonesia pada Media Sosial Twitter menggunakan Metode LSTM dan Word2Vec,” 2023. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [27] Shuvayanti Das, Sohan Ghosh, and Shubham Bhattacharya, *Proceedings of 2nd International Conference on Innovations in Electronics, Signal Processing and Communication - IESC 2019: 1st-2nd March 2019*. IEEE, 2019. Accessed: Sep. 15, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/8902427>
- [28] N. R. Ramadhanti and S. Mariyah, “Document Similarity Detection Using Indonesian Language Word2vec Model.”
- [29] C.-C. Chiu *et al.*, “State-of-the-art Speech Recognition With Sequence-to-Sequence Models,” Dec. 2017, [Online]. Available: <http://arxiv.org/abs/1712.01769>
- [30] J. Cheng, L. Dong, and M. Lapata, “Long Short-Term Memory-Networks for Machine Reading,” Jan. 2016, [Online]. Available: <http://arxiv.org/abs/1601.06733>
- [31] M. R. Faisal and D. T. Nugrahadi, “Belajar Data Science: Klasifikasi dengan Bahasa Pemrograman R,” 2019. [Online]. Available: <https://www.researchgate.net/publication/312160783>