JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 5, No. 2 March 2023 DOI: https://doi.org/10.34288/jri.v5i2.207
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Sentiment Analysis of Pedulilindungi Application Reviews Using Machine Learning and Deep Learning

Ahmad Rais Dwijaya[1], Arif Dwi Laksito[2]

Program Studi Informatika, Fakultas Ilmu Komputer
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
ahmad. dwijaya@students. amikom. ac. id[1], arif. laksito@amikom. ac. id[2]
ahmad. dwijaya@students. amikom. ac. id[1], arif. laksito@amikom. ac. id[2]
(*) Corresponding Author

Abstract

The COVID-19 pandemic that hit the world at the end of early 2020 caused many losses. The Indonesian government has established various ways to reduce the path of the COVID-19 pandemic by launching the PeduliLindungi application to reduce the spread of COVID-19. Various layers of society responded to the launch of the application with various opinions. This research mainly analyzes public opinion sentiment toward the PeduliLindungi application, as determined by 10,000 reviews on the Google Play Store. This study aims to compare the performance of deep learning and machine learning models in sentiment analysis. The stages of the research method begin with data collection methods, data pre-processing, and sentiment analysis using a machine learning model with the embedding of the word TF-IDF, which includes the Nave Bayes algorithm, Decision Tree, Random Forest, K-Nearest Neighbour, and SVM. As for the deep learning model with the fastText word embedding word representation technique using the LSTM algorithm, an evaluation is carried out using the confusion matrix. The results of this study state that deep learning models perform better than machine learning models.

Keywords: Sentiment Analysis; Machine Learning; Deep Learning; LSTM

Abstrak

Pandemi covid-19 yang melanda dunia pada akhir awal 2020 menimbulkan banyak kerugian. Pemerintah Indonesia menetapkan berbagai cara untuk mengurangi dampak dari pandemi covid-19 dengan meluncurkan aplikasi PeduliLindungi sebagai alat untuk mengurangi penyebaran covid-19. Berbagai lapisan masyarakat menanggapi peluncuran aplikasi tersebut dengan berbagai opini. Analisis sentimen opini publik terhadap aplikasi PeduliLindungi yang ditentukan dari review di Google Play Store sebanyak 10.000 review menjadi fokus utama penelitian ini. Tujuan dari penelitian ini adalah membandingkan performa dari model Deep Learning dan Machine Learning dalam melakukan analisis sentiment. Tahapan metode penelitian diawali dengan metode pengumpulan data, pre-processing data, dan analisis sentiment menggunakan model machine learning dengan penyematan kata TF-IDF yang mencakup algoritma Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor dan SVM. Sedangkan untuk model Deep Learning dengan teknik representasi kata Fasttext Word Embedding menggunakan algoritma LSTM kemudian dilakukan evaluasi menggunakan Confusion Matrix. Hasil dari penelitian ini menyatakan bahwa model deep learning mempunyai kinerja yang lebih baik dibandingkan model machine learning.

Kata kunci: Analisis Sentimen; Machine Learning; Deep Learning; LSTM

INTRODUCTION

In the midst of the rapid development of various fields in the world, in 2020, the world was shocked by the outbreak of a new virus that spread quickly. Indonesia was also attacked by a new virus called Coronavirus Disease 2019 (COVID-19) which is caused by a strain of coronavirus (SARS-CoV2) (Djalante et al., 2020). The 2019 coronavirus is an infectious illness brought on by a brand-new virus that has never been identified. This COVID-19 illness can spread from person to person and has flu-like symptoms (Sudiarsa & Wiraditya, 2020). WHO and the Indonesian government both declared Covid-19 to be a disease that causes public health emergencies and non-natural disasters. (Keputusan Menteri Kesehatan Republik Indonesia, 2020).

A new achievement for the government of the Republic of Indonesia in utilizing technology is the availability of the PeduliLindungi application, which is expected to inhibit the spread of the Covid-19 virus. Community involvement is required for this software to operate (Sudiarsa & Wiraditya, 2020). A feature of the PeduliLindungi application allows for displaying information on vaccinations, test results for Covid-19, statistical data for Covid-19 cases, and telehealth providers.

The Government of the Republic of Indonesia made a policy to facilitate tracing, tracking, and fencing by requiring several places to attach a QR-Code at the entrance to check in through the PeduliLindungi application. People must install the PeduliLindungi application on their devices. It is required for people in Indonesia to install the PeduliLindungi application on the Google Play platform or similar platforms. (Pribadi, Manongga, Purnomo, Setyawan, & Hendry, 2022).

Community perspectives vary in dealing with government policies regarding the PeduliLindungi application. People's perceptions of using the PeduliLindungi application vary, as observed on the Google Play Store platform. As time passes, some users of the PeduliLindungi application reveal to support it, but others despise it (Pribadi et al., 2022). Users can voice their opinions in various ways, from subtly complimenting phrases to bluntly disparaging ones. Users can also provide reviews of downloaded applications in the form of star ratings (between 1 and 5).

This study's main objective is to observe user comments on the PeduliLindungi app on the Google Play Store website. Many customers have left positive and negative reviews regarding complaints and other opinions. Based on this, sentiment analysis of the PeduliLindungi application on the Google Play Store platform is needed because it will become test data for application developers, especially the Ministry of Communication and Informatics, to improve application performance.

Sentiment analysis is a method for automatically classifying several texts into positive or negative attitudes (Dashtipour, Gogate, Adeel, Larijani, & Hussain, 2021). The lexicon-based approach and the machine learning approach are two methods for sentiment analysis. (Onan, 2021). However, deep learning models can also be used to build classification models for sentiment analysis.

Several studies have used machine learning models for sentiment analysis, such as the K-Nearest Neighbours (KNN), Naive Bayes, Support Vector Machine (SVM), Random Forest, and Naive Bayes algorithms. Some studies compare the many methods of machine learning models as well as those that use only one of these algorithms, as in research (Deho, Agangiba, Aryeh, & Ansah, 2018) and (Stephenie, Warsito, & Prahutama, 2020) which only conducted research using random forest algorithm.

SVM, Random Forest, and Decision Tree algorithms have been used in the study (Steinke, Wier, Simon, & Seetan, 2022). Other researchers also used Naive Bayes, SVM, decision trees, random forests, and KNN (Tran, Nguyen, & Dao, 2022). The Support Vector Machine Approach received the highest accuracy in these two studies compared to other machine learning algorithms.

In addition to employing machine learning models, several researchers have begun using deep learning models with Convolutional Neural Networks (CNN), Short-Term Long Memory (LSTM), and Recurrent Neural Network (RNN) algorithms to carry out sentiment analysis. Previous studies by (Feizollah, Ainin, Anuar, Abdullah, & Hazim, 2019; Kilimci, 2020; Kilimci & Akyokus, 2019; Ombabi, Ouarda, & Alimi, 2020) compared the performance of various sentiment analysis techniques using deep learning models. Moreover, an experiment by (Feizollah et al., 2019) obtained maximum accuracy in conducting analysis using a combination of CNN and LSTM. (Kilimci & Akyokus, 2019; Ombabi et al., 2020) They claimed that the LSTM algorithm performs the best. However, using fastText word embedding to the models (Kilimci, 2020; Ombabi et al., 2020) concluded that sentiment analysis performed better.

Recent research found that not all deep learning models are equally successful in various situations (Kapočiūtė-Dzikienė, Damaševičius, & Woźniak, 2019). Evaluated the Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB) models with the LSTM and CNN models by conducting a sentiment analysis of the opinions expressed in Lithuania on the news portal Lietuvos Ryta. The highest accuracy results were obtained for classification using machine learning, namely MNB and SVM. Deep learning with word2vec and FasText works less than optimally, so the accuracy is not as good as machine learning.

This study aims to compare the effectiveness of machine learning and deep learning models to identify the most effective model for sentiment analysis on PeduliLindungi application reviews on the Google Play Store.

RESEARCH METHODS

In this study, we compared the performance of machine learning models, which include Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbours, and SVM, with the TF-IDF word embedding technique and LSTM deep learning model using fastText word embedding. The dataset was obtained from a scraping review of the PeduliLindungi application on the Google Play Store and then evaluated to measure the performance. The framework of this research is shown in Figure 1.

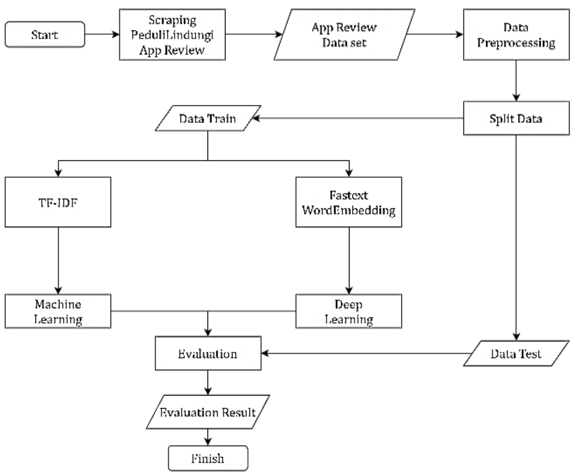

Figure 1. Research Framework

Scraping PeduliLindungi App Review

Scraping PeduliLindungi App review data from the Google Play Store uses Python programming, utilizing the google-play-scrapper library with parameters sort = sort.NEWEST means the latest data. Country = ‘id’, which means a review from Indonesia. Lang = ‘id,’ which means review in Indonesian, and count = 10000, which indicates the amount of data taken, as many as 10,000. Based on the required data, it is filtered and then assigned a sentiment label based on the review rating. Ratings 1 and 2 are categorized as having negative sentiment, ratings 4 and 5 have positive sentiment, and rating 3 has neutral sentiment.

Pre-processing Data

The data pre-processing process is carried out after collecting and labeling. At this stage, the data is prepared for analysis (Pribadi et al., 2022). The pre-processing data stage consists of several sub-processes: symbol/punctuation/emoji removal, case folding, tokenization, filtering, and stemming. During case folding, all letters are reduced to lowercase. Then in the tokenization stage, the strings (text) sequence is split into a keyword, word, phrase or other element called a token (Dey et al., 2020). Words that occur frequently but have meaning in the analysis or stop words are removed during filtering. The stemming stage involves removing affixes or suffixes (both at the beginning and the end of a word) to get to the root word. The Stop Word Remover Factory package from Sastrawi is used in the last two processes.

Split Dataset

Data splitting refers to the division of data into two or more parts. This is a crucial machine learning component, especially for building data-driven models. Typically, a two-part split is used to train the model, while the other part tests or evaluates the data. In this work, we used the train test split function of the Scikit-Learn package to split the complete dataset into 80% training data and 20% test data.

TF-IDF

To see the response from the text you have, each word will be weighted with specific rules. The authors used the TF-IDF (Term Frequency — Inverse Document Frequency) method for word embedding in this study. This method calculates the value of the term frequency (TF) and inverse document frequency (IDF) for each token in the document in the corpus. In simple terms, the TF-IDF method determines the number of times a word appears in a document. TF-IDF involves multiplying the IDF's size by the TF's size, which has proven to be very strong compared to other models (Robertson, 2004).

The fastText Word Embedding

Word embedding is a method of representing words as solid vectors that captures the relationships or semantic similarity between words that appear in later paragraphs of the text. (Mikolov, Grave, Bojanowski, Puhrsch, & Joulin, 2018). Proposed fastText, an enhanced word embedding method with sub-word insertion (n-gram characters). FastText represents words as minor elements using n-gram characters. Each word is divided into n-gram characters where $3 \leq n \leq 6$. For example, with n=3, the word “smart” is then divided into <sm, sma, mar, art, dan re>, and <smart>. FastText also uses the skip-gram approach with negative samples suggested for Word2Vec's modified skip-gram loss function.

Machine Learning

The author uses traditional machine learning techniques, such as Naive Bayes, Decision Tree, Random Forest, K-Nearest Neighbours (KNN), and Support Vector Machine (SVM), to analyze sentiment in this study. The conditional probability model serves as the basis for the Naive Bayes classifier. Sentiment classification involves two vectors, so the classifier assumes the two features' independence and probabilities (Zahoor, Bawany, &

P-ISSN: 2656-1743 | E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v5i2.207

Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Hamid, 2020). A decision tree is a hierarchical model for supervised learning in which the decision nodes of the test function identify a local region as a recursive set of subdivisions (Bayhaqy, Sfenrianto, Nainggolan, & Kaburuan, 2018). Random Forest works in two steps. The first step combines N decision trees to create a random forest. Then the second step is to make predictions for each tree made in the first step(Pribadi et al., 2022). KNN is a classification algorithm that forms new data classes using the closest K data (neighbors) as a guide (Bayhaqy et al., 2018). The SVM algorithm separates class data by finding the most optimal hyperlink (Firmansyah, Asnawi, Hasanah, Novian, & Pravitasari, 2021).

LSTM

This study uses deep learning models, especially Long Short-Term Memory (LSTM), to analyze sentiment. Researchers have widely explored the LSTM method, producing findings superior to previous methods, making it an ideal way to apply sentiment analysis. (Romadhoni, Fahmi, & Holle, 2022).

LSTM was developed to infer remote dependencies in sequence data. Long-term dependencies between data are maintained within the LSTM and contain semantic context. This algorithm uses special cells or storage units to store information about dependencies in the remote context. Each LSTM unit contains input, forget, and output gates to control the information stored, forgotten, and passed on to the next step. LSTM units decide what to store and when to allow reading, writing, and erasing through gate bypassing or blocking information through LSTM units (Kilimci & Akyokus, 2019). The network architecture is shown in Figure 2.

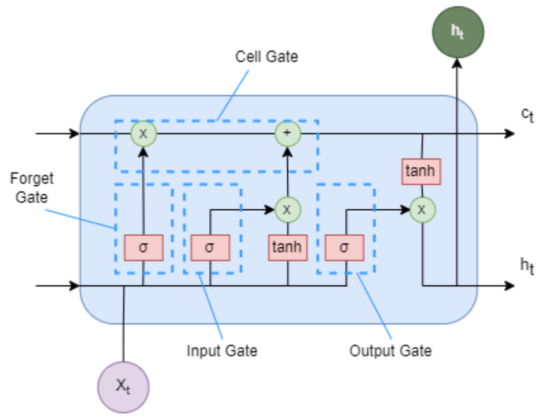

Figure 2. LSTM Network Cell

The four gate units that make up an LSTM are the input gate, forget gate, cell gate, and output gate (Romadhoni et al., 2022). Gates' job is determining whether the information should be kept or forgotten. Equation 1 to 6 shows the calculation for the hidden layer in the LSTM cell.

$f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right)$ .... (1)

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

$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$ (1) .... (3)

$\tilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$ (1)............ (4)

$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ .... (5)

$h_t = o_t * \tanh(c_t)$ .... (6)

Information :

f_t = Forget gate

i_t = Input Gate

c_t = Cell Gate

o_t = Output Gate

h_t = Hidden State

$\tilde{c}_t$ = Intermediate cell state

b = Bias

W = Weight

Evaluation

The last part of this experiment is measuring sentiment analysis performance utilizing deep learning and machine learning, and then the confusion matrix is required to calculate the accuracy or performance value. Equations 7 to 10 describe the confusion matrix.

$Precision = \frac{TP}{TP+FP} \times 100\%$............ (7)

$recall = \frac{TP}{TP+FN} \times 100\%$ .... (8)

$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$ .... (9)

$F1 - Score = 2 \times \frac{precision \times recall}{precision+recall} \times 100\%$ .... (10)

Information :

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

This experiment's last stage compares machine learning and deep learning performance. Accuracy and f1-score from the confusion matrix are used as matrix evaluation from both approaches.

RESULTS AND DISCUSSION

The authors employed Python version 3.10 to do the experiments, which ran on a Windows environment with an AMD Ryzen 5 2.5Ghz processor and 8Gb of RAM. The dataset used is 10,000 reviews of the PeduliLindungi application from the Google Play Store platform. Each review takes information about the content of the review (content) and the rating given by the reviewer (score), which is used for the labeling process. Data labeling in this study is limited to rating review. Ratings 1 and 2 are grouped into negative sentiments, while ratings 4 and 5 are grouped into positive sentiments, as shown in Table 1.

Table 1. Datasets Example

content	score	sentiment
Semakin kesini semakin berat aplikasinya,..buka Aplikasi lama bgt..Tolong diperbaiki bug utk scan barcode! 🙏🙏	2	Negative
Sangat membantu setiap melakukan perjalanan. Walaupun kadang lemot prosesnya	4	Positive
Mau download sertifikat internasional aja ga bisa ,ga muncul hasil nya, bikin ribet.	1	Negative
Terima kasih kepada pemerintah yang telah memperhatikan masyarakat dengan aplikasi ini	5	Positive

Because of less relevant for the rating 3 content, we ignore this. So, the total data become 9,477 containing positive and negative sentiment; the composition can be seen in figure 3.

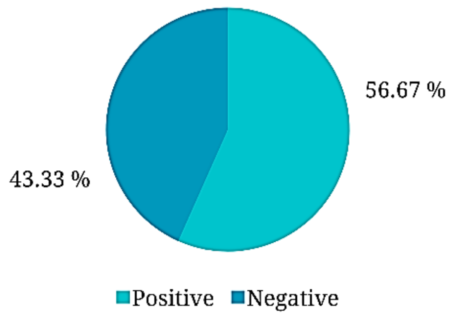

Figure 3. Percentage of The Amount of Data

The example of several stages of pre-processing text describes in table 2 below.

Table 2. Data Pre-processing Example

Data Review	Semakin kesini semakin berat aplikasinya,..buka Aplikasi lama bgt..Tolong diperbaiki bug utk scan barcode! 🙏🙏
Removal of symbols/punctuation marks/emojis	Semakin kesini semakin berat aplikasinya buka Aplikasi lama bgt Tolong diperbaiki bug utk scan barcode
Case folding	semakin kesini semakin berat aplikasinya buka aplikasi lama bgt tolong diperbaiki bug utk scan barcode
Tokenises	[Semakin, kesini, semakin, berat, aplikasinya, buka, aplikasi, lama, bgt, tolong, diperbaiki, bug, utk, scan, barcode]
Filtering	[Semakin, kesini, semakin, berat, aplikasinya, buka, aplikasi, lama, tolong, diperbaiki, bug, scan, barcode]
Stemming	makin sini makin berat aplikasi buka aplikasi lama tolong baik bug scan barcode

P-ISSN: 2656-1743 | E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v5i2.207

Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

The dataset is divided into two parts with a ratio of 80:20. 20% of the dataset is used as a data test, and the remaining 80% as a data train. Before splitting the datasets, random oversampling (ROS) is used to overcome imbalanced data. In a machine learning model experiment, the training data is feature extracted using TF-IDF to calculate each word's TF and IDF scores. Then classification is carried out using algorithms from machine learning models, namely Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbors (KNN) with a value of K = 10, and Support Vector Machine (SVM) with parameters kernel = linear and C = 5. Each algorithm takes less than 1 minute to classify.

To evaluate, calculations are performed using recall, precision, and accuracy models based on the confusion matrix. The results of evaluation testing using the confusion matrix from experiments using machine learning models found that the Decision Tree and SVM algorithms perform better than the others with an accuracy of 84.5%, as in table 3.

Table 3. Machine Learning Model Evaluation Results

Algorithm		Precision	Recall	F1-score	Accuracy
Naïve Bayes	positive	83.6%	86.7%	85.1%	82,8 %
	negative	81.5%	77.5%	79.5%	
Decision Tree	positive	94.5%	77.2%	85.1%	84,5 %
	negative	75.7%	94.1%	83.9%	
Random Forest	positive	94.2%	77.5%	85.1%	84,3 %
	negative	75.8%	93.5%	83.7%	
K-Nearest Neighbors	positive	79.7%	77.3%	78.5%	83,1 %
	negative	74.5%	92.4%	82.4%	
Support Vector Machine	positive	94.5%	77.2%	85.1%	84,5 %
	negative	75.7%	94.1%	83.9%	

In addition to machine learning models, deep learning models were also used in the experiments. A deep learning experiment required 8 minutes and 2 seconds to load the 4.2 GB of trained Indonesian word vector data for the training data's word embedding using fastText Word Embedding. Then classification is carried out using the deep learning model algorithm, namely LSTM, with the parameter batch size = 64 and the number of epochs = 200. The LSTM technique requires 15 minutes and 34 seconds to train a classification model. An accuracy of 87% is obtained for the classification results by utilizing the LSTM 1 layer, as shown in Table 4.

Table 4. Deep Learning Model Evaluation Results

Algorithm	LSTM 1 Layer	
	positive	negative
Precision	85 %	89 %
Recall	90 %	84 %
F1-Score	87 %	86 %
Accuracy	87 %	

Figures 4 and 5 show the model accuracy and loss of the estimated training and validation data sets in the LSTM network units.

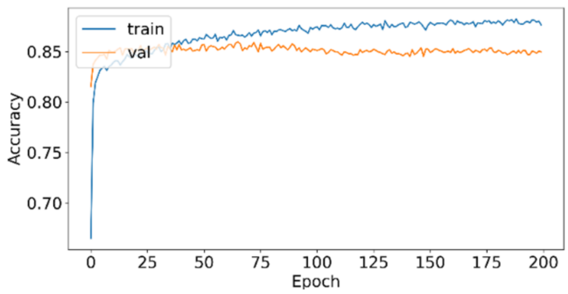

Figure 4. Model accuracy and validation using the LSTM network

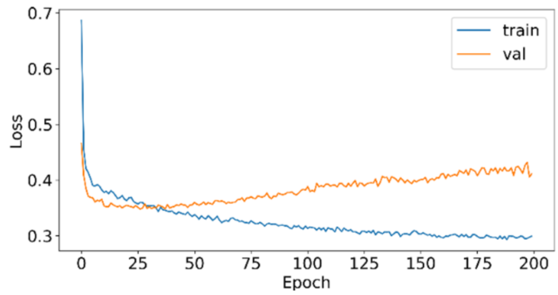

Figure 5. Loss accuracy and validation using the LSTM network

CONCLUSION AND SUGGESTIONS

In this study, we tested the effectiveness of the machine learning and deep learning models for sentiment analysis of the PeduliLindungi application review. A total of 9,479 reviews were used. Positive reviews reached 56.67% of the total reviews, while negative reviews reached 43.33%.

The results show that compared to the machine learning model that uses TF-IDF word embedding, the deep learning model that combines the LSTM algorithm with the fastText Word Embedding word representation technique has the best performance and the highest accuracy.

It is noticeable that we are labeling the dataset with the help of a rating review. However, it appears that we could not control the review's sentiment. The upcoming studies should consider the labeling method. Moreover, comparing recurrent neural network methods like Gate Recurrent Unit (GRU) and LSTM will be challenging.

REFERENCE

Bayhaqy, A., Sfenrianto, S., Nainggolan, K., & Kaburuan, E. R. (2018). Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes. *2018 International Conference on Orange Technologies, ICOT 2018*, 1–6. https://doi.org/10.1109/ICOT.2018.8705796

Dashtipour, K., Gogate, M., Adeel, A., Larijani, H., & Hussain, A. (2021). Sentiment analysis of persian movie reviews using deep learning. *Entropy*, *23*(5), 1–16. https://doi.org/10.3390/e23050596

Deho, O. B., Agangiba, W. A., Aryeh, F. L., & Ansah, J. A. (2018). Sentiment analysis with word embedding. *IEEE International Conference on Adaptive Science and Technology, ICAST*, *2018-Augus*(August), 1–4. https://doi.org/10.1109/ICASTECH.2018.8506717

Dey, S., Wasif, S., Tonmoy, D. S., Sultana, S., Sarkar, J., & Dey, M. (2020). A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews. *2020 International Conference on Contemporary Computing and Applications, IC3A 2020*, 217–220. https://doi.org/10.1109/IC3A48958.2020.233300

Djalante, R., Lassa, J., Setiamarga, D., Sudjatma, A., Indrawan, M., Haryanto, B., … Warsilah, H. (2020). Review and analysis of current responses to COVID-19 in Indonesia: Period of January to March 2020. *Progress in Disaster Science*, *6*. https://doi.org/10.1016/j.pdisas.2020.100091

Feizollah, A., Ainin, S., Anuar, N. B., Abdullah, N. A. B., & Hazim, M. (2019). Halal Products on Twitter: Data Extraction and Sentiment Analysis Using Stack of Deep Learning Algorithms. *IEEE Access*, *7*, 83354–83362. https://doi.org/10.1109/ACCESS.2019.2923275

Firmansyah, I., Asnawi, M. H., Hasanah, S. A., Novian, R., & Pravitasari, A. A. (2021). A Comparison of Support Vector Machine and Naïve Bayes Classifier in Binary Sentiment Reviews for PeduliLindungi Application. *2021 International Conference on Artificial Intelligence and Big Data Analytics, ICAIBDA 2021*, (18), 140–145. https://doi.org/10.1109/ICAIBDA53487.2021.9689771

Kapočiūtė-Dzikienė, J., Damaševičius, R., & Woźniak, M. (2019). Sentiment analysis of Lithuanian texts using traditional and deep learning approaches. *Computers*, *8*(1). https://doi.org/10.3390/computers8010004

Keputusan Menteri Kesehatan Republik Indonesia. (2020). Keputusan Menteri Kesehatan Republik Indonesia Nomor HK.01.07/MenKes/413/2020 Tentang Pedoman Pencegahan dan Pengendalian Corona Virus Disease 2019 (Covid-19). *MenKes/413/2020*, *2019*, 207.

Kilimci, Z. H. (2020). Sentiment analysis based direction prediction in bitcoin using deep learning algorithms and word embedding models. *International Journal of Intelligent Systems and Applications in Engineering*, *8*(2), 60–65. https://doi.org/10.18201/ijisae.2020261585

Kilimci, Z. H., & Akyokus, S. (2019). The Evaluation of Word Embedding Models and Deep Learning Algorithms for Turkish Text Classification. *UBMK 2019 - Proceedings, 4th International Conference on Computer Science and Engineering*, 548–553. https://doi.org/10.1109/UBMK.2019.8907027

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 52–55. Miyazaki: European Language Resources Association (ELRA). Retrieved from https://aclanthology.org/L18-1008

Ombabi, A. H., Ouarda, W., & Alimi, A. M. (2020). Deep learning CNN–LSTM framework for

Arabic sentiment analysis using textual information shared in social networks. *Social Network Analysis and Mining*, *10*(1), 1–13. https://doi.org/10.1007/s13278-020-00668-1

Onan, A. (2021). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience*, *33*(23), 1–12. https://doi.org/10.1002/cpe.5909

Pribadi, M. R., Manongga, D., Purnomo, H. D., Setyawan, I., & Hendry. (2022). Sentiment Analysis of the PeduliLindungi on Google Play using the Random Forest Algorithm with SMOTE. *2022 International Seminar on Intelligent Technology and Its Applications: Advanced Innovations of Electrical Systems for Humanity, ISITIA 2022 - Proceeding*, 115–119. https://doi.org/10.1109/ISITIA56226.2022.9855372

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, *60*(5), 503–520. https://doi.org/10.1108/00220410410560582

Romadhoni, Y., Fahmi, K., & Holle, H. (2022). Analisis Sentimen Terhadap PERMENDIKBUD No.30 pada Media Sosial Twitter Menggunakan Metode Naive Bayes dan LSTM. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, *7*(2), 118–124.

Steinke, I., Wier, J., Simon, L., & Seetan, R. (2022). Sentiment Analysis of Online Movie Reviews using Machine Learning. *International Journal of Advanced Computer Science and Applications*, *13*(9), 618–624. https://doi.org/10.14569/IJACSA.2022.0130973

Stephenie, Warsito, B., & Prahutama, A. (2020). Sentiment Analysis on Tokopedia Product Online Reviews Using Random Forest Method. *E3S Web of Conferences*, *202*, 1–10. https://doi.org/10.1051/e3sconf/202020216006

Sudiarsa, I. W., & Wiraditya, I. G. B. (2020). Analisis Usability Pada Aplikasi Peduli Lindungi Sebagai Aplikasi Informasi Dan Tracking Covid-19 Dengan Heuristic Evaluation. *INTECOMS: Journal of Information Technology and Computer Science*, *3*(2), 354–364. https://doi.org/10.31539/intecoms.v3i2.1901

Tran, D. D., Nguyen, T. T. S., & Dao, T. H. C. (2022). Sentiment Analysis of Movie Reviews Using Machine Learning Techniques. In *Lecture Notes in Networks and Systems* (Vol. 235). Springer Singapore. https://doi.org/10.1007/978-981-16-2377-6_34

Zahoor, K., Bawany, N. Z., & Hamid, S. (2020). Sentiment analysis and classification of restaurant reviews using machine learning. *Proceedings - 2020 21st International Arab Conference on Information Technology, ACIT 2020*. https://doi.org/10.1109/ACIT50332.2020.9300098