Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

Sentiment Analysis of Wondr by BNI App Reviews on Play Store using the CNN-LSTM Method

Ihsanudin Pradana Putra*, Yuliant Sibaroni, Sri Suryani Prasetiyowati

School of Informatics, Informatics, Telkom University, Bandung, Indonesia
Email: [1,*]ihsanudinpradana@student.telkomuniversity.ac.id, [2]yuliant@telkomuniversity.ac.id, [3]srisuryani@telkomuniversity.ac.id
Correspondence Author Email: ihsanudinpradana@gmail.com
Submitted: **31/05/2025**; Accepted: **22/06/2025**; Published**: 23/06/2025**

Abstract−As the use of digital applications in banking services increases, user opinions about these applications become an important source of data to study Wondr by BNI, which receives many user reviews, is one of the applications studied in this research. This research aims to build an accurate sentiment classification model and compare the effectiveness of two word representation methods, Word2Vec and FastText, to automatically classify sentiment into two classes, positive and negative, from unstructured review text using informal language. The data was processed through pre-processing, labeling, and processing stages using a hybrid CNN-LSTM model with 20,000 reviews available on the Google Play Store. The training process is carried out using 5-fold cross-validation and hyperparameter optimization using the random search method. The results show that the model with FastText has an accuracy of 86.38%, precision of 86.82%, recall of 86.46%, and F1-score of 86.46%. In contrast, the model with Word2Vec has an accuracy of 85.90%, precision of 86.38%, recall of 85.80%, and F1-score of 85.87%. These results show that FastText is better in accuracy and performance consistency compared to Word2Vec. This research provides a better understanding of how word representation methods affect sentiment analysis in app reviews and is expected to be a reference for future development of similar models.

Keywords: Sentiment Analysis, Wondr by BNI, CNN-LSTM, Word2Vec, FastText

1. INTRODUCTION

The Industrial Revolution 4.0 has increased the number of internet users in Indonesia [1]. Based on data collected by the Indonesian Internet Service Providers Association (APJII), the number of internet users in Indonesia in 2024 reached 221.5 million people, with a penetration rate of 79.5% of the total population of 278.7 million people [2]. Digital services such as mobile banking depend on the internet [3]. In response to the global trend of banking digitalization, Bank Negara Indonesia (BNI) launched the Wondr by BNI application to meet customers' needs for fast and easily accessible services. The app has various innovative features that help users manage their finances easily through their mobile devices [4]. According to download data on the Google Play Store, the Wondr by BNI application has been downloaded 1 million times as of October 2024, with a rating of 3.4.

App ratings are usually accompanied by user reviews, which indicate the user's experience with the app such as convenience, security, and transaction speed. Reviews on the Google Play Store are quite numerous and unstructured, so a method is needed to find out how user reviews of the application [5]. Sentiment analysis on review data is very important because this text data processing method not only processes the text but also reveals the information contained in it. It is able to identify and divide user responses into positive and negative responses. The results of the analysis can be used to repair and improve service quality [6].

There has been a lot of research on sentiment analysis, such as research related to sentiment analysis on e-commercer application using CNN method getting an accuracy result of 86,6% [7]. Research related to cellular device user optimization using the CNN-LSTM hybrid model obtained an accuracy result of 92% [8]. In the banking sector, a comparison of Logistic Regression, Random Forest, Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Naive Bayes methods was conducted. LSTM achieved the best accuracy rate of 91%, SVM was second with an accuracy rate of 89% [9]. The Long Short-Term Memory method was used to review mutual fund investment applications with 99.3% accuracy [10]. The Zoom application, which uses the Convulational Neural Network (CNN) method, also uses review data from the Google Play Store, with an accuracy result of 91.5% [11].

Sentiment analysis in Indonesian is not as easy as in English. This is due to abbreviations, informal language usage, morphological complexity, and language mixtures often found in user reviews. Text pre-processing and pre-processing stages are essential to obtain good model performance due to such issues. Therefore, to improve the classification model input, word representation techniques such as Word2Vec and FastText are used. These word representation techniques have the ability to collect contextual meaning and sub-word related information.

Based on previous research, the author in this study will use the CNN method combined with LSTM. This combination was chosen because LSTM has a better ability to handle data with long sequences and maintain important temporal context [12], while CNN is effective in extracting important features from review text [13]. In [8], the feature extraction process has not been applied, so the potential accuracy improvement cannot be fully utilized. In addition, previous studies still have gaps because they do not use word representation techniques such as Word2Vec and FastText. These techniques can enrich text features and enhance the model's ability in sentiment classification. Therefore, to improve data representation quality, this study will employ the Word2Vec and FastText word representation methods. It is anticipated that model performance will be enhanced by using these methods. However, the final results depend on the amount of data, class balance, and language variation in the reviews

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

This study aims to assess the effect of feature extraction scenarios with hyperparameter optimization on the performance of the CNN-LSTM model. In addition, this study also evaluates the accuracy of the model in classifying user reviews into positive and negative sentiment categories. This research aims to provide a more complete picture of how effective the incorporation of Word2Vec and FastText is in improving the representation of text data in sentiment classification tasks. In addition, the goal of this research is to find the optimal hyperparameter configuration to improve the accuracy of the CNN-LSTM model. Therefore, the findings of this study can be used as a reference for future research in the field of natural language processing and the development of deep learning models for sentiment analysis.

2. RESEARCH METHODOLOGY

2.1 Research Stages

In this research, a system consisting of several stages is built to perform sentiment analysis on user reviews of the Wondr by BNI application. The overall system flow design can be seen in Figure 1.

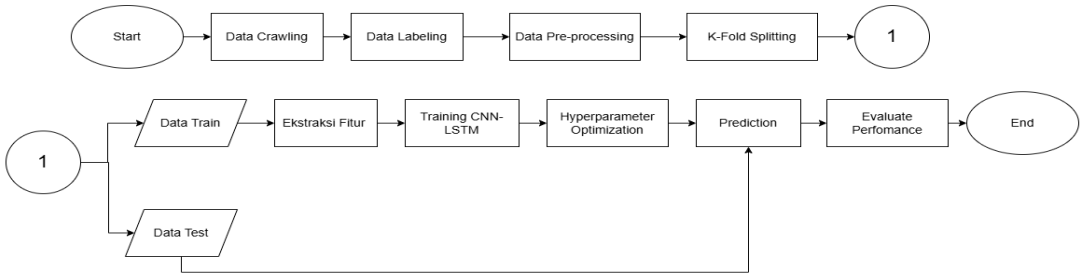

Figure 1. Flowchart of Sentiment Classification using CNN-LSTM

The system starts with user review data collected from Wondr by BNI app. After that, the data is labeled and put into two categories: positive and negative emotions. After the labeling process, a pre-processing stage is performed to clean and prepare the data. Then, the data is processed using the k-fold cross-validation technique to divide it into multiple training and testing subsets simultaneously, which ensures more reliable model evaluation and reduces bias.

The training data is then split through the feature extraction stage by incorporating Word2Vec or FastText methods. Sentiment analysis can be implemented and trained using the CNN-LSTM model. After that, a hyperparameter optimization process is used to find the most ideal combination of parameters. Next, the model is trained thoroughly using the most suitable parameters for each fold.

After the training process on each fold is completed, the model is tested on validation data to predict the sentiment of the reviews. The predicted results are compared with the actual labels using evaluation metrics such as accuracy, precision, recall, and F1 score to assess the overall performance of the model.

2.2 Data Crawling

The Python programming language was used on the Google Collaboratory platform to collect data for this study. Public reviews from Wondr by BNI app were extracted, using keywords and filters appropriate to the research objectives. In this study, 20,000 app reviews were collected over a period ranging from July 2024 to February 2025. The web scraping method was used to collect data, which extracted user reviews from BNI's Wondr application. Python programming language was used on the Google Collaboratory platform. To be used in sentiment analysis, all collected data was saved in CSV format. Table 1 shows a summary of the scraped data.

Table 1. Data Collection

Content	Rating
Uinya seger bagus, tpi masalah utamanya adalah ketika sudah terdaftar tetapi tidak bisa login lagi, Selalu nyangkut sesi login anda berakhir silahkan ulang, Sudah hapus aplikasi instal lagi tetap nyangkut disitu, tolong pengembang aplikasinya agar diperhatikan bug ini. Terima kasih	1
Cara daftarnya g ribet. Tampilan aplikasinya juga fresh. Memudahkan di era yg serba digital. Untuk qris.nya jga sudah ada. Banyak menu2 yang bermanfaat. Terakhir download bni mobile aga berat di hp ini tidak	5

2.3 Data Labelling

At this stage, each Wondr by BNI review collected is labeled. A rule-based method is used to label products based on user ratings. This process generates a dataset of 20,000 reviews, consisting of 12,029 positive reviews and 7,971 negative reviews, with ratings of 3 to 5 considered positive sentiment and ratings of 1 and 2 considered negative sentiment. The data is saved in .csv file format after labeling is complete, as shown in Table 2.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

Table 2. Data Labelling

Review	Label
Aplikasinya sangat mempermudah saat transaksi dimana pun jadi lebih praktis . Tiba2 token listrik , pulsa habis atau mau beli tiket whoos yg harus cepat pun ini sangat mudah langsung sat.. Set bayar semua tanpa kendala	1
suka banget sama tampilannya,semua fiturnya bagus dan ga ribet melakukan transaksi. mempermudah transfer ,E commerce dan top up e-wallet,bisa untuk bayar token2 & yg lainnya.pokoknya the best deh ðŸ‘ □	1
Error, bug. Setiap masuk app selalu Syarat dan ketentuan, meskipun sudah di ceklis malah sesi telah berakhir. Dicoba masuk kembali tetap sama,jd tdk bisa masuk menu. Lebih mending mbanking yg lama	0
ini gimana ya, sekarang aplikasi nya udah ga nyaman banget, riwayat transaksi ga langsung muncul, dan munculnya itu lama banget, tadi udah transfer eh mlh bukti transfer nya ga keluar, enak m-banking BNI mobile yg dulu ga kebanyak eror kayak sekarang.	0

2.4 Data Pre-Processing

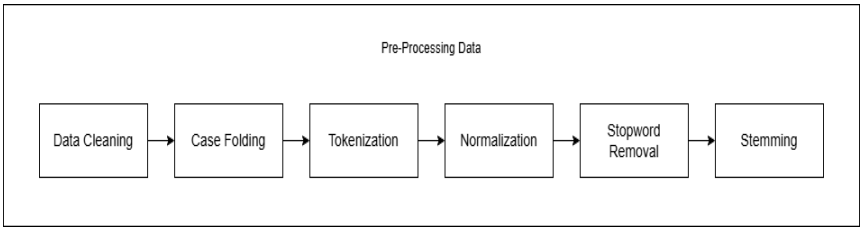

Figure 2. Data Pre-Processing

Pre-processing aims to clean and change data that is not needed during the classification process to get better results. In this research, there are several stages of pre-processing, such as:

a. Data Cleaning
Data cleaning is performed to remove punctuation marks, symbols, numbers, URLs, and irrelevant HTML tags from review text. Regular expressions from the Python re library are used to perform this procedure. With this method, any elements that do not support analysis, such as links, numbers, and special characters, can be automatically removed. In addition, cleaning also means removing unnecessary spaces to make the text format more consistent. Each of these steps makes the data more organized, clean, and ready for further analysis. Examples of punctuation, symbols and numbers in table 3.

Table 3. Data Cleaning

(.) , (,) , (?) , (!) , (;) , (:) , (-) , (--) , (=), (8..9) ,(<..=) , (/) , ((..)) , ([..]) , (`) , (~), (@) , (#) , ($) , (^) , (&) , (*) , (_) , (+), ({..}) . (|) , (>) , (<) , (1,2,3,4,5,6,7,8,9)

b. Case Folding
Case folding is done by converting all letters in the text into lowercase letters to homogenize the data. Lowercase letters to homogenize the data. This process helps make analysis easier.

c. Tokenization
Tokenization breaks text into small units of words or tokens, usually by utilizing spaces as separators. This step aims to simplify the process of analyzing or classifying text data.

d. Normalization
Text normalization involves the recognition of redundant words and replacement of words according to the KBBI. To be more accurate, this process this process is done using a manually compiled word dictionary and using literary institutions. For example, the word 'sgt' becomes ‘sangat’

e. Stopward Removal
Words that are considered unimportant or common are discarded when removing stopwords. Although these words usually have a grammatical function, they do not provide useful data for analyzing text. Overall, data analysis is more effective with their removal. This study uses a combination of the default stopword list in Indonesian from the NLTK library and a custom stopword list loaded from a file called stopwordbahasa.csv to provide a more comprehensive filtering process tailored to the context of the dataset being studied.

f. Stemming
The stemming process simplifies the text for further analysis by converting a compound word into a base word by removing the the affix.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

Before sentiment analysis is carried out, the data that has been collected must go through a preprocessing stage to clean it up so that it is ready to be used in further analysis. The results of the preprocessing process are shown in Table 4.

Table 4. Pre-Processing Data

Steps	Review
App Review	Nah suruh migrasi tapi belum siap, tarik tunai ATM, pembayaran menggunakan QRIS, isi E-wallet dengan nominal diatas 2juta juga tidak bisa, tampilan juga lebih simple yang BNI mobile banking.. Terlalu banyak menu jendela untuk 1 menu. Perbaiki segera buat susah kami pengguna!
Data Cleaning	Nah suruh migrasi tapi belum siap tarik tunai ATM pembayaran menggunakan QRIS isi Ewallet dengan nominal diatas juta juga tidak bisa tampilan juga lebih simple yang BNI mobile banking Terlalu banyak menu jendela untuk menu Perbaiki segera buat susah kami pengguna
Case Folding	nah suruh migrasi tapi belum siap tarik tunai atm pembayaran menggunakan qris isi ewallet dengan nominal diatas juta juga tidak bisa tampilan juga lebih simple yang bni mobile banking terlalu banyak menu jendela untuk menu perbaiki segera buat susah kami pengguna
Tokenization	['nah', 'suruh', 'migrasi', 'tapi', 'belum', 'siap', 'tarik', 'tunai', 'atm', 'pembayaran', 'menggunakan', 'qris', 'isi', 'ewallet', 'dengan', 'nominal', 'diatas', 'juta', 'juga', 'tidak', 'bisa', 'tampilan', 'juga', 'lebih', 'simple', 'yang', 'bni', 'mobile', 'banking', 'terlalu', 'banyak', 'menu', 'jendela', 'untuk', 'menu', 'perbaiki', 'segera', 'buat', 'susah', 'kami', 'pengguna']
Normalization	['nah', 'suruh', 'migrasi', 'tapi', 'belum', 'siap', 'tarik', 'tunai', 'atm', 'pembayaran', 'menggunakan', 'qris', 'isi', 'ewallet', 'dengan', 'nominal', 'diatas', 'juta', 'juga', 'tidak', 'bisa', 'tampilan', 'juga', 'lebih', 'simple', 'yang', 'bni', 'mobile', 'banking', 'terlalu', 'banyak', 'menu', 'jendela', 'untuk', 'menu', 'perbaiki', 'segera', 'buat', 'susah', 'kami', 'pengguna']
Stopword Removal	['suruh', 'migrasi', 'tarik', 'tunai', 'atm', 'pembayaran', 'qris', 'isi', 'ewallet', 'nominal', 'diatas', 'juta', 'tampilan', 'simple', 'bni', 'mobile', 'banking', 'menu', 'jendela', 'menu', 'perbaiki', 'susah', 'pengguna']
Stemming	suruh migrasi tarik tunai atm bayar qris isi ewallet nominal atas juta tampil simple bni mobile banking menu jendela menu baik susah guna

Table 4 shows that the pretreatment procedure was successful in making the app review data clearer and more suited for analysis. The reviews were initially presented as unstructured phrases with a variety of symbols, unstandardized terms, and messy constructions. Irrelevant items, including symbols or numbers, are eliminated during the data cleaning stage. For uniformity, the case folding procedure then changes all of the letters to lowercase. After tokenization, which separates the text into individual words, non-standard terms are normalized to take on a conventional form. Stopword removal is then used to eliminate frequent terms like "yang," "dan," and "untuk" that don't have any significant meaning. Last but not least, stemming reduces a word to its most basic form, such as "pembayaran" to "bayar." The end outcome demonstrates that the review content gets more focused, succinct, and suitable for analysis in the subsequent sentiment step. After all preprocessing is completed, the number of results that are analyzed is around 19.817 data. The data's label sentiment is composed of 11.859 positive (label 1) and 7.958 negative (label 0).

2.5 Data Splitting

To ensure that the model is evaluated thoroughly, a five-time data division scheme is used to perform cross-validation, meaning that the entire data is used alternately as training and validation data. Each combination of hyperparameters tested through the random search process is evaluated using this division scheme to ensure that the model works in the best possible way.

2.6 Word2Vec

Word2Vec is a technique for displaying words in the form of fixed-dimensional numerical vectors that indicate their semantic meaning according to the context in which they are used. The two main architectures of Word2Vec are Continuous Bag-of-Words (CBOW) and Skip-gram. CBOW predicts the context around the target word, while Skip-gram predicts the context of the target word. Sentiment analysis, text classification, and machine translation are some of the natural language processing tasks that often use these representations [14], [15].

2.7 FastText

FastText breaks words into n-grams to generate word representations that consider morphological information. Unlike Word2Vec, which processes each word, FastText captures the internal structure of words such as prefixes and suffixes, which makes it better at displaying rare or absent words in the vocabulary. FastText uses the Skip-gram architecture to learn embedding based on the context of a collection of n-grams [16].

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

2.8 CNN-LSTM

A combined model intended to address the problem of analyzing data containing both spatial and temporal elements, such as text, images, or videos. This combination utilizes the ability of CNN to extract important features from input data, such as patterns in images or text [17], and LSTM to capture sequence dependencies or temporal relationships between data [18]. The following is an overview of the process of CNN-LSTM.

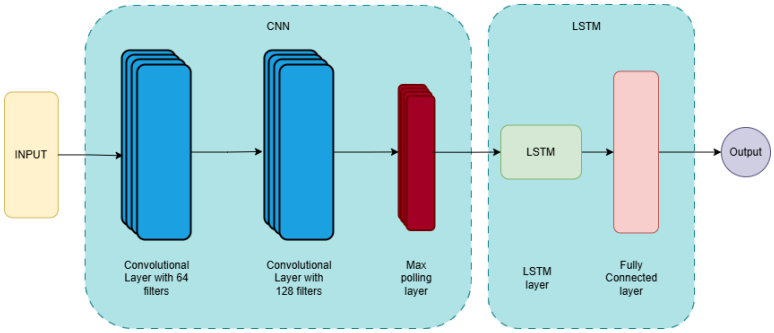

Figure 3. CNN-LSTM Model Illustration

This research utilizes a CNN-LSTM hybrid architecture for sentiment classification on user reviews of the Wondr by BNI app. Using a pre-trained and static Word2Vec or FastText embedding, the input data consists of a 100-token long word sequence mapped to a 300-dimensional vector. The 15,000 most frequently occurring words are included in this embedding. Next, the data is processed through two convolutional layers successively, with the number of filters and kernel size determined through hyperparameter search; the number of filters tested at each layer is 64, 128, and 256, and the kernel size is 3, 4, or 5. To reduce dimensionality, ReLU activation function is used at each convolutional layer, and then max pooling of size 2 is applied. Padding is used by default, meaning "valid", and the step is not explicitly set (default=1).

To obtain sequential information from the text, the pooling results are passed to the LSTM layer. The number of LSTM units ranges from 64, 128, or 256, depending on the hyperparameter search results. The return_sequences parameter is not enabled, so only the last output is used. To avoid overfitting, a dropout layer with a rate of 0.3-0.5 was added. According to the number of classes and softmax activation function, the output of the LSTM is directed to a dense layer with the number of neurons. The model is compiled using Adam's optimizer and a categorical crossentropy loss function. Training is performed in a maximum number of ten epochs with batch sizes of 32, 64, or 128. If there was no improvement in validation, training was stopped early with patience 3 [19].

2.9 Hyperparameter Optimization

Proper hyperparameter values are critical to machine learning as small variations in hyperparameter values can lead to significant differences in model performance. While incorrect settings can lead to overfitting or underfitting, an ideal hyperparameter combination can improve the accuracy and generalization of the model.

During the optimization process, the random search method is used to find the best hyperparameter combination in the CNN-LSTM model. This method randomly selects a combination of values from the hyperparameter search space, including the number of LSTM units, CNN kernel size, and dropout rate. Each iteration involves training the CNN-LSTM model on training data and evaluating test data using the randomly selected hyperparameter combination [20].

2.10 Evaluation Model

Confusion matrix is used to evaluate the performance of the classification model. This is done by comparing the actual data with the model's predicted results.

Table 5. Evaluation Model

Class	Prediction	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

For a more in-depth analysis of the model's performance, the matrix helps in calculating evaluations such as accuracy, precission, recall, and F1- score

a. Accuracy
   Accuracy is calculated by comparing the proportion of correct predictions (both true positives and true negatives) to the total number of data tested. This metric shows how well the model classifies the data as a whole. The following is the formula for accuracy:

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

b. Precission
Precision is the number of correct positive predictions among all positive predictions made by the model. It shows how many of the positive predictions are expected to be correct. The precision formula is:

$$Precission = \frac{TP}{(TP+FP)} \quad (2)$$

c. Recall
Recall reevaluates the model's ability to find all the positive data that actually exists. The number of positive data successfully recognized by the model is indicated by this metric. The recall formula is:

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

d. F1-Score
F1-Score is the harmonic mean of precision and recall used to balance both metrics, especially on imbalanced data. The F1-Score value only increases if both precision and recall are high. The formula is:

$$F1 - Score = \frac{Precission \times Recall}{(Precission + Recall)} \quad (4)$$

3. RESULT AND DISCUSSION

3.1 WordCloud Visualization Results

The experiment used word cloud visualization to analyze text data from user reviews. The text was displayed as a collection of words, or cloud, with the most frequently occurring words given a larger font size and the least frequently occurring words given a smaller font size.

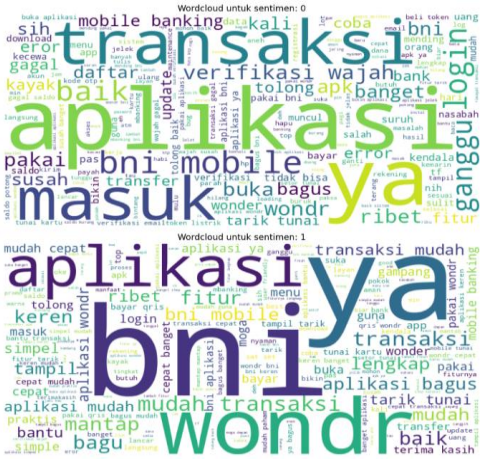

Figure 4. WordCloud Visualization

It is evident from a word cloud with the label sentimen negatif (label 0) that words like "masuk," "gagal," "verifikasi," "error," "susah," and "ribet" indicate that most users have issues related to login difficulties, wajah verification procedures, or transaction errors. Conversely, the word cloud for the positive sentiment label (label 1) was composed of words like "mudah," "fitur," "cepat," "praktis," "mantap," and "bagus," which indicate that users are concerned about the application's functionality, transaction speed, and ease of use. This visualization provides a general overview of the focus and perception of the majority of users about the Wondr application by BNI, which then serves as a basis for the classification process.

3.2 Word2Vec Results with Hyperparameters

The CNN-LSTM model was trained using Word2Vec text representations for sentiment classification of user reviews of the Wondr by BNI app. In this study, Word2Vec embeddings were constructed with a dimension size of 300, and the most frequently occurring vocabulary in the training data consisted of 15,000 tokens. Before final training, a random search method was used to perform the hyperparameter search process, which means randomly selecting parameter combinations from a predefined parameter space. This procedure was combined with a five-fold cross-validation method to ensure that the obtained parameters had stable and optimal performance across various data subsets. The optimal hyperparameter configuration was found based on these validation results, which are shown in Table 6.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

Table 6. Best Hyperparameters Word2Vec

Parameters	Value
Filters1	[128]
Filters2	[128]
Kernel_size	[5]
LSTM_unit	[256]
Dropout_rate	[0.3]
Learning_rate	[0.0005]
Batch_size	[64]
Optimizer	Adam

The best parameter configuration was used for the final training, which included up to 15 epochs. However, at the 9th epoch, the training was stopped as an early stopping mechanism with a patience value of 3 was applied to prevent overfitting. The model was compiled using Adam's optimizer and categorical cross-entropy loss function. The sentiment label format has been converted into one-hot encoding form. Figure 5 shows the training results.

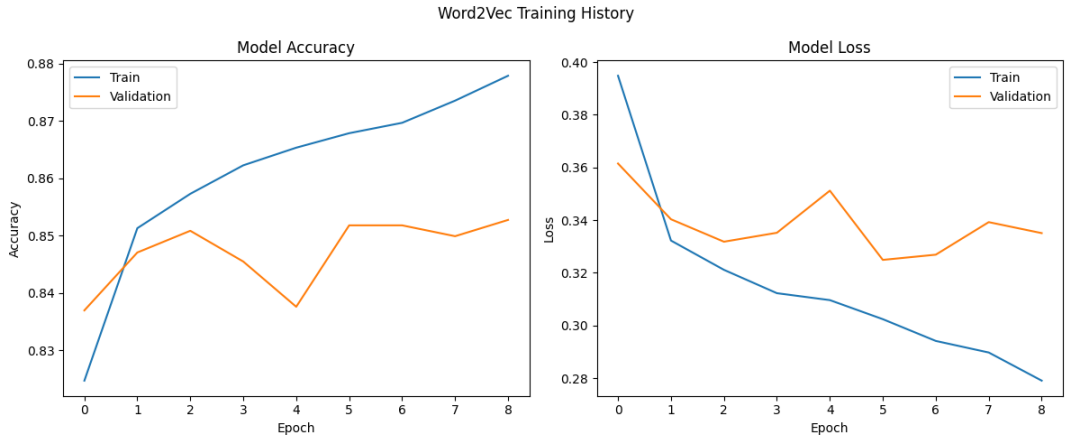

Figure 5. Word2Vec Training History

The accuracy graph shows that the training data accuracy increases consistently until it reaches about 87.8%, and the validation accuracy tends to stabilize at 85-85.7%. This indicates that the model does not suffer from significant overfitting. However, there is a slight change in the validation accuracy towards the end of the training. The loss values on the training data showed a consistent downward trend, while the validation loss values briefly changed but stabilized again. This pattern shows that the models learn well without significant overfitting, and they are able to generalize quite well to new data.

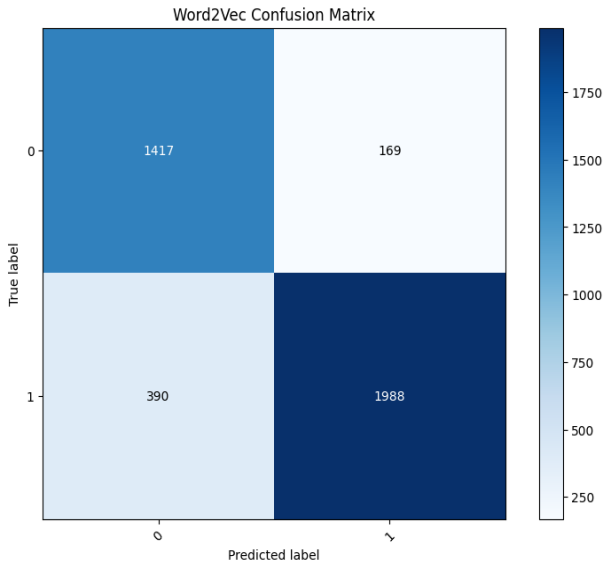

Figure 6. Confusion Matrix Word2Vec

Figure 6 shows the confusion matrix. The model classified 1,417 negative and 1,988 positive data correctly, but 169 negative data were incorrectly considered positive and 390 positive data were incorrectly considered negative.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

```
Evaluasi model Word2Vec...
62/62 [==============================] - 9s 134ms/step
Word2Vec Accuracy: 0.8590
Word2Vec Precision: 0.8666
Word2Vec Recall: 0.8590
Word2Vec F1-score: 0.8601

Word2Vec Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.89      0.84      1586
           1       0.92      0.84      0.88      2378

    accuracy                           0.86      3964
   macro avg       0.85      0.86      0.86      3964
weighted avg       0.87      0.86      0.86      3964
```

Figure 7. Classification Report Word2Vec

According to the test data evaluation, the model shows an accuracy of 85.9%, a precision of 86.66%, a recall of 85.9%, and an F1-score of 86.01%. For negative sentiment (label 0), the model shows a precision of 78% and a recall of 89%, and for positive sentiment (label 1), it shows a precision of 92% and a recall of 84%, which indicates that the model has sufficient performance.

3.3 FastText Results with Hyperparameters

In sentiment classification of user reviews of the Wondr by BNI app, the CNN-LSTM model was trained using FastText text representation to compare its performance with the Word2Vec-based model. A random search process combined with a five-step cross-validation method was used to generate stable and ideal hyperparameter configurations on various data subsets. The 15,000 most frequently occurring words in the training data were used in the FastText word representation, which was built with a vector dimension of 300. The final model was trained for a maximum of 15 epochs. However, at epoch 9 the training was stopped through an early delay mechanism with a patience value of 3 to prevent overfitting. The model was compiled using Adam's optimizer and the categorical cross-entropy loss function. This function is suitable for encryption of sentiment labels in one-hot format. Table 7 shows the validation results, which indicate the optimal hyperparameter configuration.

Table 7. Best Hyperparameters FastText

Parameters	Value
Filters1	[128]
Filters2	[128]
Kernel_size	[5]
LSTM_unit	[256]
Dropout_rate	[0.3]
Learning_rate	[0.0005]
Batch_size	[64]
Optimizer	Adam

Figure 8. FastText Training History

Figure 8 shows the training results. The accuracy graph shows a steady increase in the training data to about 87.2%, while the validation accuracy varies from 85.6% to 85.6%, indicating that the model performance remains stable during training. The loss graph shows a significant decrease in the training data, from about 0.39 to 0.28, while the loss in the validation data also shows a downward trend although punctuated by small fluctuations.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

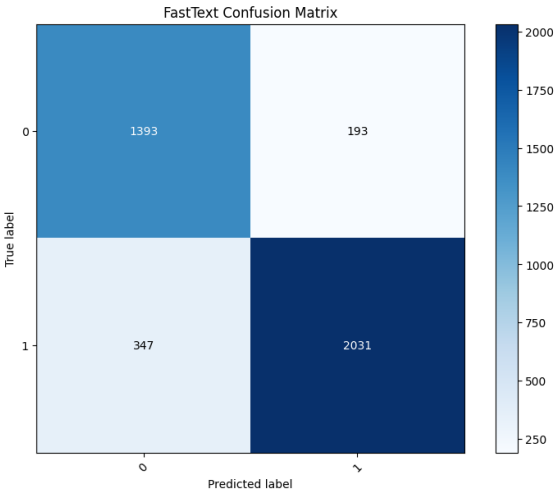

Figure 9. Confusion Matrix FastText

In Figure 9, the confusion matrix shows that the model correctly classified 1,393 negative data and 2,031 positive data. In contrast, 193 negative data were incorrectly classified as positive and 347 positive data were incorrectly classified as negative. These numbers show that the model has a good class classification balance.

```
Evaluasi model FastText...
62/62 [==============================] - 7s 109ms/step
FastText Accuracy: 0.8638
FastText Precision: 0.8682
FastText Recall: 0.8638
FastText F1-score: 0.8646

FastText Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.88      0.84      1586
           1       0.91      0.85      0.88      2378

    accuracy                           0.86      3964
   macro avg       0.86      0.87      0.86      3964
weighted avg       0.87      0.86      0.86      3964
```

Figure 10. Classification Report FastText

According to the test data evaluation, the CNN-LSTM model with FastText showed an accuracy of 86.38%, a precision of 86.82%, and a recall of 86.46%. According to the classification report, the model showed an accuracy of 80% and recall of 88% for negative sentiment (label 0), and a precision of 91% and recall of 85% for positive sentiment (label 1), which resulted in F1-score of 84% and 88%, respectively.

3.4 Comparison of Word2Vec and FastText

Training and evaluation processes have been carried out on both CNN-LSTM models using Word2Vec and FastText text representations. Both models perform well, with accuracy exceeding 85%, but there are some points that differentiate them.

Table 8. Comparison of Word2Vex and FastText

Matrix	Word2Vec	FastText
Accuracy	85.90%	86.38%
Precision	86.66%	86.82%
Recall	85.90%	86.38%
F1-Score	86.01%	86.46%

Based on the confusion matrix, the FastText model also produces a lower number of misclassifications than Word2Vec. In addition, the FastText model can correctly classify more positive data, and has greater accuracy for both sentiment labels.

FastText strong point is the ability to process sub-words (n-grams), which allows it to handle non-standard words, typos, or new words that are not in the training vocabulary. This is particularly beneficial as app review data often contains spelling variations, abbreviations, or unofficial terms. Conversely, the structural information of words is often overlooked as Word2Vec only treats words as whole units. These results indicate that FastText offers a richer

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

and more flexible word representation, which will ultimately have a positive impact on the classification accuracy of the model, although the performance difference between the two models is not very large. Consequently, it can be concluded that the combination of CNN-LSTM with FastText is the most efficient for performing sentiment analysis of app reviews in this study.

4. CONCLUSION

Based on the results of previous research, this study compares the performance of the CNN-LSTM model with two word representation methods, Word2Vec and FastText, in performing sentiment analysis on user reviews of the Wondr by BNI application available on the Google Play Store. The feature extraction process and hyperparameter optimization contribute to the improvement of model performance. The CNN-LSTM model with FastText achieved the highest accuracy of 86.38%, slightly higher than Word2Vec, which reached 85.90%. Evaluation of accuracy, recall, and F1 score metrics shows that both models have the ability to classify sentiment equally between positive and negative categories. The model hyperparameters were obtained through a random search process combined with 5 fold cross-validation and early stopping to prevent overfitting during training. Although the model shows good results, this study has some limitations. First, the data used only comes from one platform, Google Play Store, so it does not convey a wider range of user opinions. Secondly, the optimization process only uses random search, which although effective, does not include a thorough examination of parameter combinations. Thirdly, this study only compares two word representation approaches. Therefore, further research is recommended to collect data from various sources, use other optimization techniques such as grid search, and study broader word representations such as GloVe, ELMo, or transformer-based models such as BERT. It is expected that more accurate classification results that match the complexity of user review data will be generated.

REFERENCES

[1] H. Kusuma and W. K. Asmoro, "Perkembangan Financial Technologi (Fintech) Berdasarkan Perspektif Ekonomi Islam," *Istithmar J. Studi Ekon. Syariah*, vol. 4, no. 2, Dec. 2020, doi: 10.30762/istithmar.v4i2.14.

[2] Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), "Survei Penetrasi & Profil Pengguna Internet Indonesia 2024." Accessed: Oct. 16, 2024. [Online]. Available: https://survei.apjii.or.id/home

[3] L. Abubakar and T. Handayani, "Penguatan Regulasi: Upaya Percepatan Transformasi Digital Perbankan Di Era Ekonomi DigitaL," *Masal.-Masal. Huk.*, vol. 51, no. 3, pp. 259–270, Jul. 2022, doi: 10.14710/mmh.51.3.2022.259-270.

[4] R. Hardiartama, A. A. Arifiyanti, and S. F. A. Wati, "Application of Ensemble Machine Learning Methods for Aspect-Based Sentiment Analysis on User Reviews of the Wondr by BNI App," *J. Teknol. DAN OPEN SOURCE*, vol. 8, no. 1, pp. 97–111, Jun. 2025, doi: 10.36378/jtos.v8i1.4297.

[5] S. Fransiska and A. I. Gufroni, "Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method," *Sci. J. Inform.*, vol. 7, no. 2, pp. 203–212, Nov. 2020, doi: 10.15294/sji.v7i2.25596.

[6] I. Busrayan, "Analisis Sentimen Pelanggan Terhadap Aplikasi Wondr By Bni Menggunakan Naive Bayes, Support Vector Machine (Svm), Dan K- Nearest Neighbor (KNN)," *J. Comput. Sci. Inf. Technol.*, vol. 2, no. 2, pp. 264–274, 2025.

[7] F. A. Khatami, B. Irawan, and S. Si, "Analisis Sentimen Terhadap Review Aplikasi Layanan E-Commerce Menggunakan Metode Convolutional Neural Network," *E-Proceeding Eng.*, vol. 7, pp. 4559–4566, 2020.

[8] Yuhefizar, Ismael, Arif Rizki Marsa, Dedi Mardianto, and Ronal Watrianthos, "Implementasi Model Hybrid CNN-LSTM untuk Optimasi Pengalaman Pengguna Perangkat Seluler," *TEMATIK*, vol. 11, no. 2, pp. 204–212, Dec. 2024, doi: 10.38204/tematik.v11i2.2125.

[9] J. Aguirre-Sosa, M. L. Dextre, M. Lozada-Urbano, and J. A. Vargas-Merino, "Background of Peruvian gastronomy and its perspectives: an assessment of its current growth," *J. Ethn. Foods*, vol. 10, no. 1, p. 50, Dec. 2023, doi: 10.1186/s42779-023-00212-4.

[10] T. I. Hermanto and D. Irmayanti, "Sentiment Analysis of User Reviews of Mutual Fund Investment Applications on Google Playstore using Long Short Term Memory (LSTM) Algorithm," *KLIK Kaji. Ilm. Inform. Dan Komput.*, vol. 4, no. 1, pp. 200–207, Agustus 2023, doi: 10.30865/klik.v4i1.1109.

[11] R. Refianti and N. Anggraeni, "Sentiment Analysis Using Convolutional Neural Network Method to Classify Reviews on Zoom Cloud Meetings Application Based on Reviews on Google Playstore," *Int. J. Eng. Sci. Inf. Technol.*, vol. 3, no. 3, pp. 7–16, Sep. 2023, doi: 10.52088/ijesty.v3i3.463.

[12] P. Alkhairi, A. P. Windarto, M.Kom, and M. M. Efendi, "Optimasi LSTM Mengurangi Overfitting untuk Klasifikasi Teks Menggunakan Kumpulan Data Ulasan Film Kaggle IMDB," *Build. Inform. Technol. Sci. BITS*, vol. 6, no. 2, Sep. 2024, doi: 10.47065/bits.v6i2.5850.

[13] T. H. Alip Maskhuri, "Analisis Sentimen Pengguna pada Aplikasi Tokopedia Menggunakan Algoritma Convolutional Neural Network," *Build. Inform. Technol. Sci. BITS*, vol. 6, no. 4, pp. 2501–2511, 2025, doi: DOI 10.47065/bits.v6i4.6923.

[14] F. W. Kurniawan and D. W. Maharani, "Analisis Sentimen Twitter Bahasa Indonesia dengan Word2Vec," *E-Proceeding Eng.*, vol. 7, pp. 7821–7828, Agustus 2020.

[15] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 8, no. 1, p. 147, Feb. 2021, doi: 10.25126/jtiik.0813944.

[16] M. A. A. Islamy, "Analisis Sentimen IMDB Movie Reviews menggunakan Metode Long Short-Term Memory dan FastText," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 6, pp. 4106–4115, Sep. 2022.

[17] A. Z. R. Adam and E. B. Setiawan, "Social Media Sentiment Analysis using Convolutional Neural Network (CNN) dan Gated Recurrent Unit (GRU)," *J. Ilm. Tek. Elektro Komput. Dan Inform. JITEKI*, vol. 9, no. 1, pp. 119–131, 2023, doi: 10.26555/jiteki.v9il.25813.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 1, June 2025 Page: 561−571
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i1.7477

[18] H. Handoko, A. Asrofiq, J. Junadhi, and A. S. Negara, "Sentiment Analysis of Sirekap Tweets Using CNN Algorithm," *INTENSIF J. Ilm. Penelit. Dan Penerapan Teknol. Sist. Inf.*, vol. 8, no. 2, pp. 312–329, Aug. 2024, doi: 10.29407/intensif.v8i2.23046.

[19] M. Ridwan and A. Muzakir, "Hate Speech Classification Model On Twitter Data Using Cnn-Lstm," *TEKNOMATIKA*, vol. 12, no. 2, pp. 209–218, 2022.

[20] A. R. Fitriansyah and Y. Sibaroni, "Analisis Sentimen Terhadap Pembangunan Kereta Cepat Jakarta - Bandung Pada Media Sosial Twitter Menggunakan Metode SVM dan GloVe Word Embedding," *E-Proceeding Eng.*, vol. 10, no. 2, pp. 1713–1723, Apr. 2023.