Indonesian Journal of Electrical Engineering and Computer Science
Vol. 38, No. 1, April 2025, pp. 677~688
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v38.i1.pp677-688

Sentiment analysis based on Indonesian language lexicon and IndoBERT on user reviews PLN mobile application

Yessy Asri[1], Dwina Kuswardani[2], Widya Nita Suliyanti[2], Yosef Owen Manullang[2], Atikah Rifdah Ansyari[1]
[1]Department of Information Systems, Faculty of Energy and Telematics, Institute Technology of PLN, Jakarta, Indonesia
[2]Department of Informatics Engineering, Faculty of Energy and Telematics, Institute Technology of PLN, Jakarta, Indonesia

Article Info

Article history:

Received Mar 14, 2024
Revised Sep 26, 2024
Accepted Nov 7, 2024

Keywords:

IndoBERT
Indonesian language lexicon
InSet lexicon
PLN mobile application
Sentiment analysis

ABSTRACT

PLN mobile application as an integrated platform for self-service among mobile consumers, facilitating easier access to various services, including receiving information such as public complaints. The application can be downloaded through the Google Play Store and App Store, and users can express their opinions through reviews and ratings. In this era of advanced technology, aspects such as reviews, ratings, and evaluations have important value for business practitioners. However, there are often inconsistencies between ratings and reviews that do not fully represent the quality of the application. In response, a study was conducted to analyze the sentiment of user reviews from January to June 2022, by collecting 1,000 review samples from the Google Play Store. The data was collected using web scraping techniques and then processed into a dataset through text pre-processing methods. Sentiments were analyzed using an automatic labeling method in Indonesian based on a lexicon known as INSET (Indonesia sentiment), which resulted in 482 positive reviews, 144 negative reviews, and 374 neutral reviews. The next step is classification using Indonesian bidirectional encoder representations from transformers (IndoBERT). In this process, the data was divided into testing, training, and validation sets with a ratio of 80:10:10. The analysis managed to achieve an impressive accuracy rate of 81%.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Yessy Asri
Department of Information Systems, Faculty of Energy and Telematics, Institute Technology PLN
Jakarta 11750, Indonesia
Email: yessyasri@itpln.ac.id

1. INTRODUCTION

The PLN mobile application serves as a digital platform intended to provide electricity-related services via mobile devices. Its array of online services encompasses bill payments, electricity credit purchases, self-meter reading submissions, power augmentation requests, complaint and disruption reporting, credit monitoring, postpaid electricity consumption monitoring, bill and power outage notifications, information on disruption resolution progress, and power network maintenance.

Within the Google Play Store [1], user reviews for the PLN mobile application [2] are assigned ratings ranging from 1 to 5. As of 2022, the PLN mobile app boasts a commendable rating of 4.7 out of 5 [3], indicating a favorable overall evaluation by its users. These reviews, generally detailed and comprehensive, provide insights into users' experiences with PLN mobile. Through their feedback, users strive to offer valuable input to application developers, aiding in feature enhancement, service quality improvement, and the provision of beneficial suggestions for other users. However, despite the provision of detailed and beneficial reviews, usersoften encounter challenges in assigning ratings that align with their written evaluations.

This incongruence occasionally leads to ratings that fail to accurately represent the application's quality. Reviews expressed in sentence format more accurately reflect user responses to the PLN mobile application, thus influencing prospective users or downloaders [4].

The disparity between assigned ratings and provided reviews highlights discord, as indicated by instances where five-star ratings are accompanied by negative reviews. This discrepancy underscores the presence of dissatisfaction despite ostensibly positive ratings. In light of this, sentiment analysis research becomes essential to comprehensively understanding user reviews. Such studies hold the potential to enhance user experiences, address identified shortcomings, and elevate overall user satisfaction. Sentiment analysis, an important aspect of natural language processing (NLP), seeks to identify, comprehend, and interpret sentiments or opinions embedded within textual data. In the context of this research, sentiment analysis relates to user reviews of the PLN mobile application.

Currently, many lexicon-based sentiment analysis labeling systems focus on the English language. Previous study [5] utilized Vader lexicon labeling. Vader lexicon is an English-based labeling system, therefore researchers must first translate review data before processing and categorizing it. The testing model employs the machine learning Naïve Bayes approach. The resulting accuracy is 70%, although this does not represent the best results because the labeling method is in English and the testing model employs machine learning.

In this study, improvements were made with the Indonesian lexicon-based labeling method and testing using deep learning using the Indonesian bidirectional encoder representations from transformers (IndoBERT) algorithm. The InSet lexicon labeling method has been tested in [6], [7]. The results of the study prove that INSET (Indonesia sentiment) has better performance compared to Vania lexicon. One of the reasons is because the Vania lexicon [8] is developed from English root words.

The IndoBERT model is used in this research because it is a deep learning pre-trained model adapted from BERT for Indonesian, developed using a transformer-based architecture, and trained on a substantial Indonesian corpus. IndoBERT excels in a variety of NLP tasks, including sentiment analysis. Its pre-training equips IndoBERT to recognize and understand contextual nuances in the analyzed text, making it well suited for sentiment analysis tasks [9]. The IndoBERT model was chosen as the model in this study due to its ability to process the complex and diverse Indonesian language. The IndoBERT model has been trained for general NLP tasks, including sentiment analysis, so it has high performance for such tasks. In addition, IndoBERT was chosen as the model for sentiment analysis because it has the advantage of processing Indonesian language and the ability to understand the context of the given text. In addition, IndoBERT is a trained natural language model specifically designed for the Indonesian language, so it has better knowledge and understanding of the Indonesian language and its context [10], [11].

In this study, the authors conducted research to test whether the use of deep learning with an Indonesian-based lexicon can improve accuracy compared to previous research or vice versa. In addition, the author wants to examine whether using the Indonesian-based lexicon method (INSET lexicon) with the IndoBERT model will result in low accuracy due to dictionary limitations or the difficulty of adding new words to a predefined dictionary or will actually increase accuracy compared to previous research. The author will also evaluate the results of the IndoBERT model in performing sentiment analysis using a special Indonesian language model. The results of this sentiment analysis process will be used to see the consistency between ratings and reviews given on PLN mobile application reviews.

2. RELATED RESEARCH

The PLN mobile application is an integrated platform for self-service facilities aimed at mobile consumers, created by PLN. The PLN mobile application facilitates easier access to various services, including receiving diverse information such as a community complaint forum, capacity enhancement, electricity billing, and more. This research focuses on the classification of user review data for the PLN mobile application, which can provide insights or input to PLN concerning the PLN mobile application. Many methods for classifying review data utilize machine learning and deep learning techniques.

The research in [11] explores sentiment towards online lectures in Indonesia using lexicon-based approaches, with particular attention to the COVID-19 pandemic's impact on online education. Leveraging the INSET lexicon, specifically tailored for Indonesian, it classified sentiments into positive, negative, and neutral categories, uncovering that 63.4% of tweets were negative, 27.6% positive, and 8.9% neutral. The application of preprocessing techniques, such as cleaning, tokenization, filtering, and stemming, played a vital role in achieving an overall accuracy of 79.2%, and an average precision of 72.9%. This research contributes to the expanding field of sentiment analysis, highlighting the importance of understanding public sentiment towards online learning and providing localized insights in the Indonesian context during a global pandemic. Have produced an Indonesian sentiment lexicon called InSet, comprising 3,609 positive words and 6,609 negative words. Each word was manually labeled based on its polarity and enhanced by adding stemming and synonyms [10]. This approach successfully outperformed all existing baseline methods, achieving the highest accuracy of 65.78%. The study suggests that lexicon enhancement can be accomplished by combining translated English lexicons with InSet. Figure 1 shows of InSet lexicon algorithm.

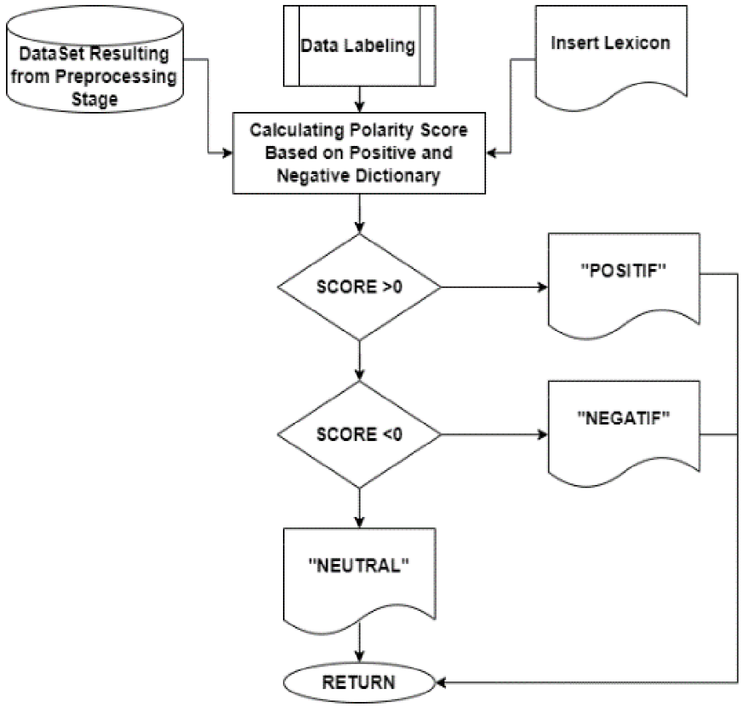


Figure 1. InSet lexicon algorithm

In the InSet lexicon's data labeling phase, every word or entry is given a distinct label or category, such as positive, negative, or neutral, used for sentiment analysis purposes. These categorizations are established by the context of word usage within applicable texts or documents. The InSet lexicon includes various Indonesian words with positive or negative attributes, each assigned a particular weight. The weights of the words fall within the range of greater than 0 and less than 0; a value under 0 denotes that the word has a negative sentiment, while a value over 0 indicates that the word has a positive sentiment [10], [12].

Researcher in [13]-[15], the paper elaborates on BERT, a transformer-based technique for NLP originally conceived by Jacob Devlin and his colleagues at Google, introduced in 2018 [14]. BERT facilitates the handling of bidirectional representation in anonymized text by effectively amalgamating context from both the left and right sections. Modest adjustments to the pre-existing BERT model can yield solutions to a myriad of challenges. BERT's strength lies in its simplicity and robust interpretability. This characteristic accounts for BERT's versatility across 11 programming languages and its impressive performance metrics, including an 80.5% general language understanding evaluation (GLUE) score, 86.7% multi-genre natural language Inference (MultiNLI) accuracy, 93.2% Stanford question answering dataset (SQuAD) v1.1 F1 test performance, and 83.1% SQuAD v2.0 F1 test performance. The standardized data outcomes are visually depicted in Figure 2 [13].

During the pre-training phase, BERT employs two unsupervised tasks, as illustrated in Figure 2. The initial process is termed masked LM, a procedure wherein the model utilizes surrounding context words to attempt to forecast the [MASK] word. The model undergoes training by designating [MASK] to a stochastic percentage of the input tokens, followed by predicting those [MASK] tokens. As mentioned in [13], masking was implemented for 15% of all randomly generated word piece tokens. A limitation of this model is the potential misalignment between the pre-training and fine-tuning stages, stemming from the absence of the [MASK] token during fine-tuning. This issue can be resolved by not consistently substituting masked words with actual [MASK] tokens; rather, 80% are substituted with [MASK] tokens, 10% with random words, and the remaining 10% are left unaltered [13].

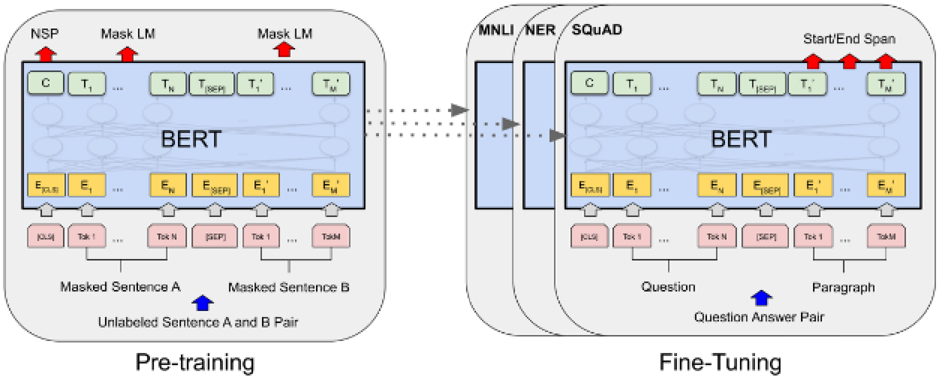


Figure 2. Pre-training and fine-tuning model BERT

The subsequent procedure is known as next sentence prediction (NSP), wherein the model is given a pair of sentences as input and is trained to determine if the second sentence follows the first in the actual document. During the training phase, as cited in [13], half of the inputs consist of pairs where the second sentence is indeed the following sentence in the original document, while the remaining half are composed of random sentences from the corpus, chosen to be the second sentence. It is presumed that the randomly selected sentence will be unrelated to the first sentence [13]. Figure 3 illustrates a depiction of the input process carried out on the BERT model, and the outcome of the standardized data is presented in Figure 3.

The research in [13]-[15], directed at demonstrating the performance achievable through the utilization of multiple tasks using the IndoBERT model. IndoBERT is an adaptation and variant of the BERT methodology specific to the Indonesian context, originally developed in 2018 by a team of researchers at Google artificial intelligence (AI) language. This model has been implemented to forecast the subsequent sentence in Google's search queries. Within the sentiment analysis task, employing the IndoBERT approach yields an F1-score metric of 84.13, a value that surpasses the results obtained by other methodologies applied to the identical dataset, including Naïve Bayes, logistic regression, bidirectional long short-term memory (BiLSTM) with fast text, MBERT, and MalayBERT [16]-[19].

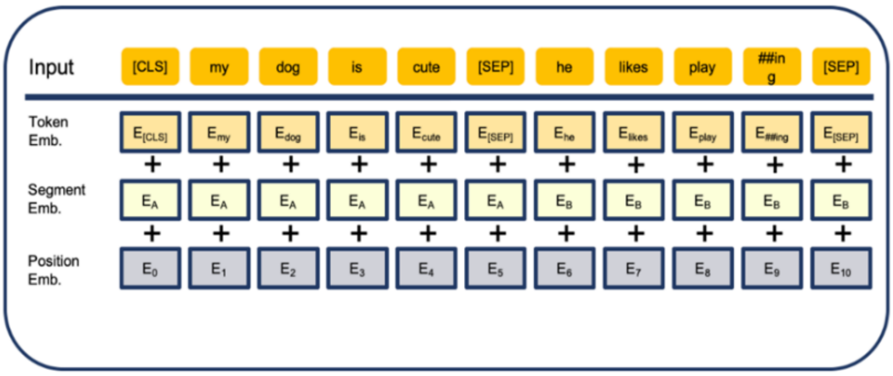


Figure 3. BERT input representation

3. METHOD

Figure 4 illustrates the workflow methodology for the PLN mobile review data from January to June 2022 using the InSet lexicon labeling and the IndoBERT model. In this research, data collection was carried out using web scraping techniques. The specific target for scraping is PLN mobile application review information related to the PLN mobile application available on Google Play Store, resulting in a dataset of 1,000 review data points from a population of 67,951 (January-June 2022) using stratified random sampling data techniques [20]. The scraping data is then filtered into four columns. The columns that will be filtered are username, at, score, and content to facilitate analysis at the next stage. Figure 5 shows of the pre-processing data workflow.

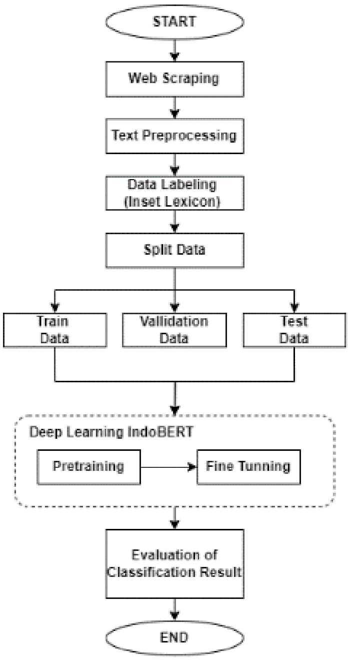


Figure 4. The research workflow

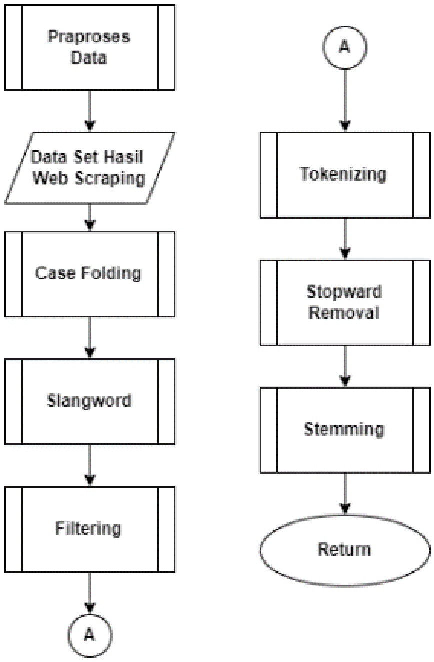


Figure 5. The pre-processing data workflow

4. RESULTS AND DISCUSSION

4.1. Text pre-processing

After obtaining data from the PLN mobile application, Figure 5 showing the data will be processed through several text-preprocessing stages, namely case folding, normalization (slang word), filtering, tokenization, stop word removal, and stemming. To carry out this text-preprocessing, the author uses the Python programming language and installs the natural language toolkit (NLTK) library, which is used to clean the data [21]-[23]. In the initial stage of the preprocessing process, namely the case folding process, regex, pandas, numpy, and regex libraries are used. This library is used to change text from uppercase to lowercase. Slang word is the stage of changing non-standard words into standard words. The author uses a slang word dictionary of 15,084 dictionaries to carry out the process of changing words in the text. The following is the source code for the slang word, as shown in Figures 6 and 7 shows of result of the slang word process.

At the filtering stage, changing the affixes from each word that has been filtered into basic words in the text will be deleted, as will removing repeated words and punctuation marks. The data has been case-folded, and then a filtering process is carried out to change the affixed words of each word that has been filtered into base words, with the text being deleted as well as the deletion of repeated words and the deletion of punctuation marks such as periods, semicolons, commas, quotation marks, numbers, italics, and so on. Figure 8 shows the source code of filtering step. In the tokenization stage, the text is changed into pieces of words consisting of one token, and then in the next stage, namely stop removal, general words that do not have a special meaning are deleted (usually in the form of conjunctions, auxiliary words, and other general words) from tokenizing. In the final stage of the data pre-processing process, namely the stemming stage, words will be converted into root words using the sastrawi library.


```
data_slang = pd.read_csv('kamus_slangword.csv')
def replace_slang(ulasan):
    res = ''
    for item in ulasan.split():
        if item in data_slang.slang.values:
            res += data_slang[data_slang['slang'] == item]['formal'].iloc[0]
        else:
            res += item
        res += ' '
    return res
```

Figure 6. Source code of the slang word

```
#slangword
df['preprocessing'] = df['preprocessing'].apply(replace_slang)
df.to_csv('replace_slang.csv')
df[['ulasan', 'preprocessing']]
```

	ulasan	preprocessing
0	Sangat membantu,urusan jadi lebih mudah. Namun...	sangat membantu,urusan jadi lebih mudah. namun...
1	Udah pengajuan menambah speed iconnet dari 10m...	sudah pengajuan menambah speed iconnet dari 10...
2	Aplikasi membingungkan. Bagaimana pelanggan ak...	aplikasi membingungkan. bagaimana pelanggan ak...
3	Kan sy mau pasang iconnet nih terus tdi teknis...	kan saya mau pasang iconnet nih terus tadi tek...
4	Kenapa pln mobil blum bisa top up saldo,kog ma...	kenapa pln mobil belum bisa bagus up saldo,kog...
...	...	...
995	Mengapa fitur data pembayaran yg sdh dilakukan...	mengapa fitur data pembayaran yang sudah dilak...
996	Sesuai motonya "semua makin mudah", Laporan Ga...	sesuai motonya "semua makin mudah", laporan ga...
997	Say kasih bintang 4 dulu,saya sangat kesulitan...	saya kasih bintang 4 dulu,saya sangat kesulita...
998	Untuk menu penyambungan baru, ketika menentuka...	untuk menu penyambungan baru, ketika menentuka...
999	Aplikasinya bagus, semua fitur2 nya juga keren...	aplikasinya bagus, semua fitur2 nya juga keren...

1000 rows × 2 columns

Figure 7. Result of the slang word process

```
import string
def filteringText(ulasan):
    ulasan = re.sub(r'@[A-Za-z0-9]+', '', ulasan)
    ulasan = re.sub(r'#[A-Za-z0-9]+', '', ulasan)
    ulasan = re.sub(r'RT[\s]', '', ulasan)
    ulasan = re.sub(r"http\S+", '', ulasan)
    ulasan = re.sub(r'[0-9]+', '', ulasan)
    ulasan = re.sub(r'(.)\1+',r'\1\1', ulasan)
    ulasan = re.sub(r'[\?\.\!]+(?=[\?.\!])', '',ulasan)
    ulasan = re.sub(r'[^a-zA-Z]',' ', ulasan)
    ulasan = re.sub(r'\b(\w+)( \1\b)+', r'\1', ulasan)
    ulasan = ulasan.replace('\n', ' ')
    ulasan = ulasan.translate(str.maketrans('', '', string.punctuation))
    ulasan = ulasan.strip(' ')
    return ulasan
```

Figure 8. Source code of filtering step

4.2. Labelling data

After the text-preprocessing stage, clean data was obtained, and then data labeling was carried out using InSet lexicon on 1,000 Indonesian language review data. The following is the source code for labeling using the InSet lexicon in Figures 9 and 10 shows of example of data results labeled data. Figure 10 is an example of data results labeled using positive and negative dictionaries. Data labeling uses an InSet lexicon, taken from the sentiment value of each word in the positive and negative dictionary, where the value for positive sentiment is obtained from a score >0, and for negative sentiment, it is obtained from a score <0. Sentiment polarity is a measure of the subjectivity and objectivity of a sentence [24], [25]. Figure 11 is the result of data labeling using InSet lexicon, which obtained 482 or 48.20% positive labels, 374 or 37.40% neutral labels, and 144 or 14.40% negative labels.

```
lexicon_positive = pd.read_excel('kamus_positif.xlsx')
lexicon_positive_dict = {}
for index, row in lexicon_positive.iterrows():
    if row[0] not in lexicon_positive_dict:
        lexicon_positive_dict[row[0]] = row[1]

lexicon_negative = pd.read_excel('kamus_negatif.xlsx')
lexicon_negative_dict = {}
for index, row in lexicon_negative.iterrows():
    if row[0] not in lexicon_negative_dict:
        lexicon_negative_dict[row[0]] = row[1]

def sentiment_analysis_lexicon_indonesia(ulasan):
    score = 0
    for word in ulasan:
        if (word in lexicon_positive_dict):
            score = score + lexicon_positive_dict[word]
    for word in ulasan:
        if (word in lexicon_negative_dict):
            score = score + lexicon_negative_dict[word]
    sentimen=''
    if (score > 0):
        sentimen = 'positif'
    elif (score < 0):
        sentimen = 'negatif'
    else:
        sentimen = 'netral'
    return score, sentimen

results = df['preprocessing'].apply(sentiment_analysis_lexicon_indonesia)
results = list(zip(*results))
df['label'] = results[0]
#data['sentimen'] = results[1]
#data

df['label'] = results[1]
dataSentimen = df
data_inset = df

data_inset[['ulasan', 'preprocessing', 'label']]
```

Figure 9. Source code of InSet lexicon

	ulasan	preprocessing	label
0	Sangat membantu,urusan jadi lebih mudah. Namun...	[membantu, urusan, mudah, , buka, aplikasinya,...	positif
1	Udah pengajuan menambah speed iconnet dari 10m...	[pengajuan, menambah, speed, iconnet, mbps, mb...	netral
2	Aplikasi membingungkan. Bagaimana pelanggan ak...	[aplikasi, membingungkan, , pelanggan, memaham...	negatif
3	Kan sy mau pasang iconnet nih terus tdi teknis...	[pasang, iconnet, nih, teknisi, dtng, pasang, ...	netral
4	Kenapa pln mobil blum bisa top up saldo,kog ma...	[pln, mobil, top, up, saldo, kog, lwat, accont...	positif
...	...	...	...
995	Mengapa fitur data pembayaran yg sdh dilakukan...	[fitur, data, pembayaran, dibanding, versi, , ...	netral
996	Sesuai motonya "semua makin mudah", Laporan Ga...	[sesuai, motonya, , mudah, , , laporan, ganggu...	positif
997	Say kasih bintang 4 dulu,saya sangat kesulitan...	[kasih, bintang, , kesulitan, masuk, fitur, ub...	negatif
998	Untuk menu penyambungan baru, ketika menentuka...	[menu, penyambungan, , menentukan, titik, loka...	netral
999	Aplikasinya bagus, semua fitur2 nya juga keren...	[aplikasinya, bagus, , fitur, nya, keren, , mu...	positif

1000 rows × 3 columns

Figure 10. Example of data results labeled data

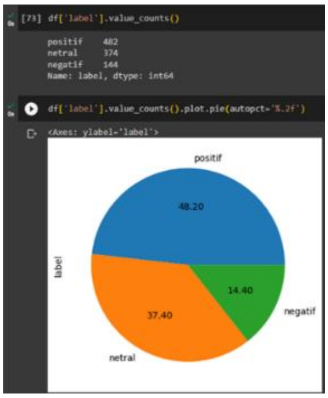


Figure 11. InSet lexicon labeling percentage

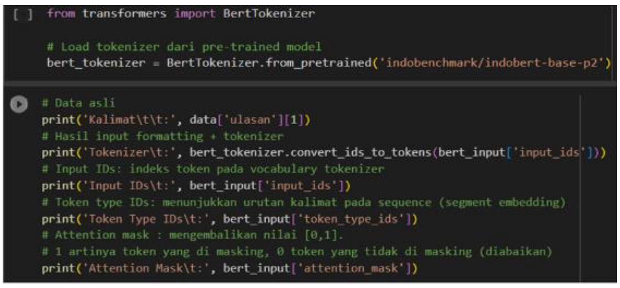
4.3. Modelling

4.3.1. Pre-trained

At this stage, the text data is converted into a representation that can be understood by the BERT model. This process produces output in the form of a dictionary (dictionary) containing tokens that have been tokenized, input IDs, attention masks, and token type IDs. The source code for the BertTokenizer process can be seen in the following Figure 12.

In Figure 12, BertTokenizer is imported from the transformer's library. Then the BERT token is taken using BertTokenizer from indobert-base-p1. Next, select one example of data to create BERT input based on the BertTokenizer that has been taken. Enter the BERT input into the sample data used with the tokens obtained. The final stage is adding a mask barrier to the data that has been tokenized by BertTokenizer. The source code for performing BertTokenizer on all the data used can be seen in Figure 13.

The processing results from BertTokenizer with an example sentence are: input "*aju tambah speed iconnet mbps mbps info gmail pesan informasi*," then the sentence undergoes a tokenization process using the appropriate IndoBERT vocabulary. After tokenization, the token [CLS] is added at the beginning of the sentence, and the token [SEP] is added at the end of the sentence. Next, each token is coded based on the vocabulary index. The [CLS] token has ID 2, the *Aju* token has ID 2116, and so on. Attention masks are used to differentiate between word token values and padding values. Padding tokens (PAD) are assigned a value of 0, while word tokens are assigned a value of 1.

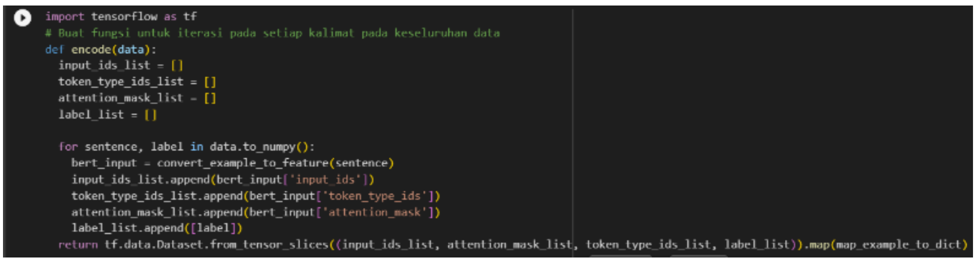


```python
from transformers import BertTokenizer

# Load tokenizer dari pre-trained model
bert_tokenizer = BertTokenizer.from_pretrained('indobenchmark/indobert-base-p2')

# Data asli
print('Kalimat\t\t:', data['ulasan'][1])
# Hasil input formatting + tokenizer
print('Tokenizer\t:', bert_tokenizer.convert_ids_to_tokens(bert_input['input_ids']))
# Input IDs: indeks token pada vocabulary tokenizer
print('Input IDs\t:', bert_input['input_ids'])
# Token type IDs: menunjukkan urutan kalimat pada sequence (segment embedding)
print('Token Type IDs\t:', bert_input['token_type_ids'])
# Attention mask : mengembalikan nilai [0,1].
# 1 artinya token yang di masking, 0 token yang tidak di masking (diabaikan)
print('Attention Mask\t:', bert_input['attention_mask'])
```

Figure 12. Source code BertTokenizer



```python
import tensorflow as tf
# Buat fungsi untuk iterasi pada setiap kalimat pada keseluruhan data
def encode(data):
  input_ids_list = []
  token_type_ids_list = []
  attention_mask_list = []
  label_list = []

  for sentence, label in data.to_numpy():
    bert_input = convert_example_to_feature(sentence)
    input_ids_list.append(bert_input['input_ids'])
    token_type_ids_list.append(bert_input['token_type_ids'])
    attention_mask_list.append(bert_input['attention_mask'])
    label_list.append([label])
  return tf.data.Dataset.from_tensor_slices((input_ids_list, attention_mask_list, token_type_ids_list, label_list)).map(map_example_to_dict)
```

Figure 13. BertTokenizer input source code

4.3.2. Fine-tunning

At this stage, classification is carried out using IndoBERT. To carry out classification by fine-tuning IndoBERT, it is done by importing the TFSert for sequence classification class from the transformer's library. This class is an implementation of the BERT model that has been set up specifically for sequence and text classification tasks. Next, load the BERT model that has been previously trained using the from_pretrained() method from the TFBert for sequence classification class. The loaded model is initialized with parameters with 'indobenchmark/indobert-base-p2, which is the Indonesian version of the BERT model with a larger language base. Also using num_labels=3 as an optional argument that determines the number of labels in the sequence or text classification task that you want to perform with the model, in this study the author used 3 labels. Next, compile the BERT model with a certain configuration using the compile() method of the model object. The hyperparameters used are learning rate with a value of 0.00003, epoch=3, and batch size=32. After that, the model produces accurate sentiment classification from testing data using encoded training data. Table 1 shows of result from IndoBERT.

Table 1. Result from IndoBERT

Epoch	Loss	Accuracy	Val_Loss	Val_accuracy
1	0.962890	0.511250	0.775613	0.660000
2	0.598350	0.760000	0.576716	0.800000
3	0.317606	0.878750	0.593226	0.790000

Table 1 displays the results of the IndoBERT model, which was trained with batch size 32, a learning rate of 0.003, and epoch 3. In the first epoch, there was a fairly high loss rate of 0.962890, while the accuracy was 0.511250. In the second epoch, the IndoBERT model produced a lower loss of 0.598350 and increased accuracy to 0.760000 compared to the previous epoch. In the third epoch, loss decreased to 0.317606 and accuracy increased to 0.878750. From the results of these three epochs, IndoBERT can be used to make accurate predictions in sentiment analysis.

4.4. Evaluation

Figure 13 depicts the evaluation outcomes of the IndoBERT prediction model, yielding an accuracy of 81%. The value of true negatives (TN) is recorded as 11, while there is 1 instance of false negatives (FN). This implies that out of a total of 100 testing data points with negative labels, IndoBERT correctly predicted 11 instances as negative, and 1 instance was incorrectly predicted, indicating that there is one data point that was predicted to be non-negative when it should have been negative (TN). Similar observations can be made for the predictions of positive and neutral labels, as indicated in Figure 14.

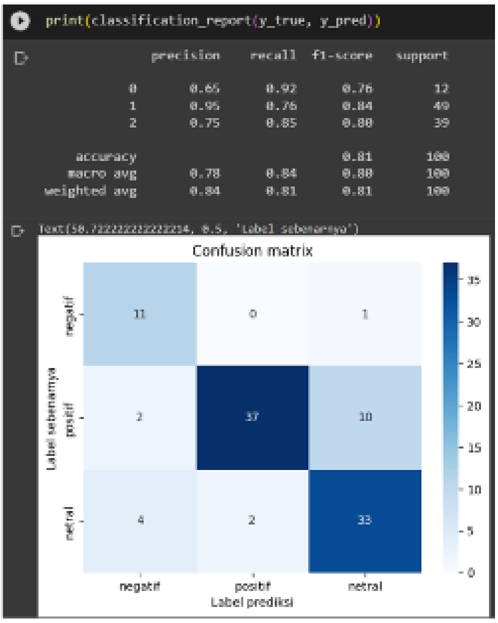


Figure 14. Evaluation with confusion matrix

In previous research researched by [5], sentiment analysis of PLN mobile application review data using the Vader lexicon automatic labeling technique and the Naïve Bayes English classification model obtained 70% accuracy, while in this study the author used data labeling using the InSet lexicon and the Indonesian language classification model IndoBERT, which got higher results than the English classification model, namely 81%. This comparison will be presented in Table 2.

Based on the introduction, it was highlighted that there exists a discrepancy between the reviews and ratings provided by users of the PLN mobile application. This incongruity was analyzed using sentiment analysis by comparing sentiment labeling using the InSet lexicon with sentiment derived from ratings. This study conducted an analysis of 1,000 reviews of the PLN mobile application, selected from a total of 67,951 reviews on the Google Play Store during the period of January to June 2022. The data preprocessing process included steps such as case folding, slangword normalization, filtering, tokenizing, stopword removal, and stemming, all labeled using the InSet lexicon [11]. Figure 15 shows out of the total analyzed reviews, 48.20% were positive (482 reviews), 37.40% were neutral (374 reviews), and the remaining 14.40% exhibited negative sentiment (144 reviews). Figure 16 shows of pie diagram from rating sentiment result.

Table 2. Comparison between IndoBERT and prior lexicon-based machine learning methods

Model	Accuracy result	Data testing	Split data	Labelling data	Labeling percentage		
					Positive	Negative	Neutral
Naïve Bayes	70%	100	90:10	Vader lexicon	47%	8%	45%
IndoBERT	81%	100	80:10:10	InSet lexicon	48.20%	14.40%	37.40%

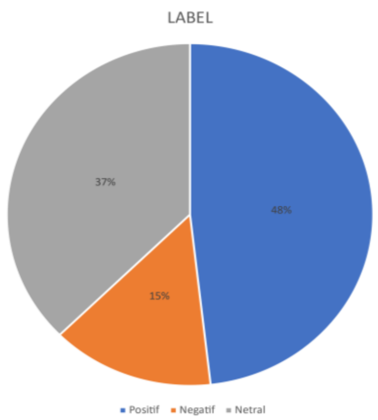


Figure 15. Pie diagram from rating sentiment

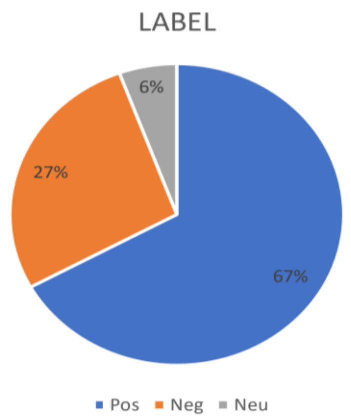


Figure 16. Result sentiment result sentiment insert lexicon

The study encompasses an analysis of 1,000 reviews of the PLN mobile application across various ratings. Out of these, 585 reviews received a rating of 5, indicating high satisfaction; 85 reviews received a rating of 4, signifying above-average satisfaction. 56 reviews with a rating of 3 displayed a neutral response. 48 reviews with a rating of 2 indicated moderate dissatisfaction, while 226 reviews with a rating of 1 reflected strong dissatisfaction. The overall average rating was 3.7. Based on these ratings, the reviews were categorized into three sentiment classes. Figure 16 shows the positive class accounted for 67% of total reviews, or 670 reviews; the neutral class encompassed 6%, or 56 reviews; and the negative class constituted 27%, or 274 reviews.

Significantly, there is a discrepancy between the ratings and sentiments expressed in the reviews. This incongruity is divided into 19% within the positive sentiment class, indicating that some reviews within this category do not entirely align with the high rating; 31% within the neutral class, highlighting a notable disparity between ratings and review text within this category; and 13% within the negative class, suggesting that certain low-rated reviews do not fully mirror negative sentiment. This analysis provides deeper insights into the dynamics between ratings and sentiments in reviews, as well as the intricate data interpretation complexity.

Training loss denotes the value obtained from calculating the loss function using the training dataset and the model's predictions. Simultaneously, validation loss represents the loss function computation using the testing dataset alongside the model's predictions, utilizing input data from the testing dataset. Analysis of such errors occurs due to discrepancies in the characteristics of these words as shown in Figure 17. Frequently occurring words in the positive sentiment class are present in the test data in Figure 17(a) (Wordcloud for sentiment positive); Figure 17(b) (Wordcloud for sentiment negative); and Figure 17(c) (Wordcloud for sentiment neutral).

(a) (b) (c)

Figure 17. Result from Worldcloud sentiment analysis for: (a) positive, (b) negative, and (c) neutral

In Figure 17(a), the distribution of frequently used or commonly appearing words in the positive sentiment dataset is presented in the form of a word cloud. This aids in identifying the frequently employed words. The words frequently appearing in positive sentiment include application, PLN, mobile, fast, easy, thank you, helpful, and excellent.

In Figure 17(b), the words that frequently emerge in the context of negative sentiment include application, PLN, electricity, disruptions, disappointment, difficulty, slowness, payments, and other similar expressions. In Figure 17(c), this visualization aids in identifying the frequently employed terms. The words frequently encountered in neutral sentiment include application, helpful, easy, payments, good, and service.

5. CONCLUSION

Analyzing the sentiment classification of user reviews on the PLN mobile application using the IndoBERT deep learning method with InSet lexicon labeling resulted in an accuracy of 81%. The accuracy produced in this study has increased by approximately 11% from previous research using the English Vader lexicon labeling method and the Naïve Bayes classification algorithm. This research shows that the accuracy results are better than previous research, but in the labeling process, there are still some reviews that are not labeled optimally. This happens because there are words that have spelling errors, so the word is not weighted correctly. One of the text preprocessing stages that plays a role in improving the performance of machine learning models is spell-checking. This is because text on the internet, especially social media, often has many typos. With so many typos, the number of vocabulary words is getting bigger. Future research will be planned for the addition of pre-processing stages, namely spelling error correction with the spelling corrector process and the construction of the electrical corpus context. The spelling corrector process detects parts of the text that are not in accordance with the correct spelling rules and produces recommendations according to the correct spelling rules. Spelling corrector work is strongly influenced by the corpus, which is used as a reference for correct word correction.

ACKNOWLEDGEMENTS

Our deepest gratitude goes to the Institute for Community Service Research (LPPM) of the PLN Institute of Technology for funding the community service activities for fiscal year 2023.

REFERENCES

[1] Q. L. Sutino and D. O. Siahaan, “Feature extraction from app reviews in google play store by considering infrequent feature and app description,” *Journal of Physics: Conference Series*, vol. 1230, no. 1, p. 012007, Jul. 2019, doi: 10.1088/1742-6596/1230/1/012007.

[2] Agung Murdifi, “PLN Relaunches PLN Mobile Application,” PLN, 2020. https://web.pln.co.id/cms/media/siaran-pers/2020/12/lebih-dekat-dengan-pelanggan-pln-luncurkan-aplikasi-new-pln-mobile/ (accessed, 2023).

[3] “PLN mobile application users,” *PLN*, 2022. https://web.pln.co.id/media/siaran-pers/2022/01/lampaui-target-2021-aplikasi-pln-mobile-diunduh-162-juta-pengguna (accessed, 2023).

[4] R. Potharaju, M. Rahman, and B. Carbunar, "A longitudinal study of Google Play," *IEEE Transactions on Computational Social Systems*, vol. 4, no. 3, pp. 135–149, Sep. 2017, doi: 10.1109/TCSS.2017.2732167.

[5] Y. Asri and M. Fajri, "Sentiment analysis of PLN mobile review data using lexicon Vader and Naive Bayes classification," in *2023 International Conference on Networking, Electrical Engineering, Computer Science, and Technology (IConNECT)*, Aug. 2023, pp. 132–137, doi: 10.1109/IConNECT56593.2023.10327064.

[6] R. Firdaus, I. Asror, and A. Herdiani, "Lexicon-based sentiment analysis of Indonesian language student feedback evaluation," *Indonesia Journal of Computing*, vol. 6, no. 1, pp. 1–12, 2021, [Online]. Available: socj.telkomuniversity.ac.id/indojc.

[7] S. Baccianella, A. Esuli, and F. Sebastiani, "SENTIWORDNET 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," in *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*, 2010, pp. 2200–2204.

[8] C. S. Khoo and S. B. Johnkhan, "Lexicon-based sentiment analysis: comparative evaluation of six sentiment lexicons," *Journal of Information Science*, vol. 44, no. 4, pp. 491–511, Aug. 2018, doi: 10.1177/0165551517703514.

[9] B. Wilie *et al.*, "IndoNLU: benchmark and resources for evaluating indonesian natural language understanding," *arXiv preprint 12009.05387*, 2020. http://arxiv.org/abs/2009.05387.

[10] F. Koto and G. Y. Rahmaningtyas, "InSet lexicon: evaluation of a word list for Indonesian sentiment analysis in microblogs," in *2017 International Conference on Asian Language Processing (IALP)*, Dec. 2017, pp. 391–394, doi: 10.1109/IALP.2017.8300625.

[11] E. W. Pamungkas and D. G. P. Putri, "An experimental study of lexicon-based sentiment analysis on Bahasa Indonesia," in *Proceedings - 2016 6th International Annual Engineering Seminar, InAES 2016*, 2016, pp. 28–31, doi: 10.1109/INAES.2016.7821901.

[12] M. Desi, K. Ulfa, E. Pradita, and S. Tri, "Sentiment analysis of online lectures in Indonesia from Twitter dataset using InSet Lexicon," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 1, pp. 24–33, 2021.

[13] R. Firdaus, I. Asror, and A. Herdiani, "Lexicon-based sentiment analysis of Indonesian language student feedback evaluation," *Indonesia Journal of Computing*, vol. 6, no. 1, pp. 1–12, 2021, doi: 10.34818/indojc.2021.6.1.408.

[14] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT 2019*, 2019, pp. 4171–4186, [Online]. Available: https://aclanthology.org/N19-1423.pdf.

[15] H. T. Madabushi, E. Kochkina, and M. Castelle, "Cost-sensitive BERT for generalisable sentence classification on imbalanced data," in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019, pp. 125–134, doi: 10.18653/v1/D19-5018.

[16] Purnomo, T. D., & Sutopo, J. (2024). "*Comparison Of Pre-Trained Bert-Based Transformer Models For Regional Language Text Sentiment Analysis In* Indonesia". *International Journal Science and Technology*, *3*(3), 11-21, doi:10.56127/ijst.v3i3.1739

[17] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: a benchmark dataset and pre-trained language model for Indonesian NLP," in *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.

[18] N. Mahfudiyah and A. Alamsyah, "Understanding user perception of ride-hailing services sentiment analysis and topic modelling using IndoBERT and BERTopic," in *2023 International Conference on Digital Business and Technology Management, ICONDBTM 2023*, Aug. 2023, pp. 1–6, doi: 10.1109/ICONDBTM59210.2023.10327320.

[19] M. F. Mubaraq and W. Maharani, "Sentiment analysis on Twitter social media towards climate change on Indonesia using IndoBERT model," *Jurnal Media Informatika Budidarma*, vol. 6, no. 4, p. 2426, 2022, doi: 10.30865/mib.v6i4.4570.

[20] I. R. Hidayat and W. Maharani, "General depression detection analysis using IndoBERT method," *International Journal on Information and Communication Technology (IJoICT)*, vol. 8, no. 1, pp. 41–51, Aug. 2022, doi: 10.21108/ijoict.v8i1.634.

[21] W. G. Cochran, *Sampling Techniques*, Third edit. New York: John Wiley & Sons, 1977.

[22] B. Juarto and Yulianto, "Indonesian news classification using IndoBert," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 2, pp. 454–460, 2023.

[23] K. T. Nwet, "Machine learning algorithms for Myanmar news classification," *International Journal on Natural Language Computing*, vol. 8, no. 4, pp. 17–24, Aug. 2019, doi: 10.5121/ijnlc.2019.8402.

[24] J. P. Haumahu, S. D. H. Permana, and Y. Yaddarabullah, "Fake news classification for Indonesian news using extreme gradient boosting (XGBoost)," *IOP Conference Series: Materials Science and Engineering*, vol. 1098, no. 5, p. 052081, Mar. 2021, doi: 10.1088/1757-899X/1098/5/052081.

[25] C. Vania, M. Ibrahim, and M. Adriani, "Sentiment lexicon generation for an under-resourced language," *International Journal of Computational Linguistics and Applications*, vol. 5, no. 1, pp. 59–72, 2014.

BIOGRAPHIES OF AUTHORS

Yessy Asri, ST., MMSI. was born on October 13, 1976, in Padang City. She starting education in the S1 Department of Informatics Engineering, then continuing his Masters education in the Department of Information Systems, and currently, she is continuing his doctoral education in the Department of Information Technology. Currently, she serves as Head of the Information Systems Study Program at the PLN Institute of Technology (ITPLN) in Jakarta and is also a lecturer at several well-known universities in Jakarta. She is also currently active as the Chief Editor of the ITPLN KILAT Journal, is active as an instructor at several training institutions, and is also currently an active BNSP assessor. With perseverance and high motivation to continue learning and trying, until now she has produced many writings in several SINTA-accredited national journals and indexed international journals on intellectual property rights (HaKI), carried out many community service activities (PkM), and received several awards. She can be contacted at email: yessyasri@itpln.ac.id.

Dr. Dwina Kuswardani M.Kom. was born in Surabaya on July 25, 1962. She graduated S1 in the Computational Mathematics Study Program, Faculty of Mathematics and Natural Sciences, University of Indonesia, in 1988. She graduated S2 in the Master of Computer Science Study Program, Faculty of Computer Science at the University of Indonesia in 2009. She graduated S3 in the Doctoral Study Program, Faculty of Computer Science, University of Indonesia, in 2016. Currently, he is a permanent lecturer in the Informatics Engineering Study Program, Faculty of Energy Telematics, Institut Teknologi PLN (ITPLN). She teaches courses in data mining, machine learning, and image processing. She can be contacted at email: dwina@itpln.ac.id.

Dr. Widya Nita Suliyanti, ST., MComps. completed her bachelor's degree in Electrical Engineering at the University Indonesia in 1996. Then he earned a Master of Computer Science degree in University of New South Wales, Australia, in 2000 and completed Doctoral education in Electrical Engineering (Computer Engineering) at the University of Indonesia in 2022. Has experience as an international journal reviewer and moderator international conferences and is an editorial member of the journal lightning. Currently a lecturer she is still in the Informatics Engineering study program at the PLN Institute of Technology. She can be contacted at email: widya@itpln.ac.id.

Yosef Owen Manullang, S.T., was born in P. Berandan on May 09, 2002. He obtained his bachelor's degree in the field of Informatics Engineering from the Faculty of Telematics Energy at the PLN Institute of Technology in 2023. Currently, he is employed as a private sector professional at GreatDay HR, a human resources company, serving as an Implementation Consultant and HR Payroll Consultant. In this capacity, his responsibilities include providing support related to payroll for client companies and educating clients on the utilization of purchased software systems and customized system features. Additionally, he engages in freelance work as a data analyst for a foreign company. He can be contacted at email: yosef1931276@itpln.ac.id.

Atikah Rifdah Ansyari was born in Barru City. On May 06, 2005. She started her education at Information Systems with awardee scholarship Aperti BUMN. Currently, she is a freelancer in marketing and admission, actively as a prompt engineer and telemarketer. She always expands experience and knowledge by innovating, being creative, having high motivation with altruist quotient, high ambition, initiative, and keen to seek new challenges. She can be contacted at email: atikah2232001@itpln.ac.id.