MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi ISSN : 1978-161X(p); 2477-2550(e)
Volume 17, No. 2(2025), pp 67-75 DOI : 10.18860/mat.v17i2. 32059
Received : Februay 8th 2025; **Accepted** : April 10th 2025 ; **Avalaible Online** : September 29th 2025

Implementation of BERT in Sentiment Analysis of National Digital Samsat (SIGNAL) User Reviews Based on Machine Learning

Ratna Savitri, Fido Rizki, and Ahmad Sobri

Abstract—**The SIGNAL application facilitates online vehicle tax payments for the public. The application's quality is frequently evaluated through user reviews on platforms like the Google Play Store. This study aims to analyze the sentiment of SIGNAL user reviews using a Machine Learning-based approach, specifically the BERT (Bidirectional Encoder Representations from Transformers) model. The dataset consists of 20,000 user reviews. After preprocessing, the remaining data comprises 17,287 reviews, categorized into 12,758 positive reviews, 2,160 neutral reviews, and 2,369 negative reviews. To address data imbalance, the Random Over Sampling (ROS) technique was applied. The evaluation was performed using metrics such as accuracy, precision, recall, and F1-score. The results of the study indicate that the IndoBERT model can classify sentiments with an accuracy of 99% and a validation accuracy of 98% after five epochs of training. Confusion matrix analysis shows that the model achieved an overall accuracy of 99.72% on training data and 98.68% on testing data. This study demonstrates that the IndoBERT model is highly effective in classifying sentiment and makes a significant contribution to understanding the user experience of SIGNAL, which can serve as a foundation for future improvements to the application.**

Index Terms— ***Sentiment Analysis, National Digital Samsat (SIGNAL), BERT, IndoBERT, Machine Learning.***

I. Introduction

In the Society 5.0 era, rapid technological advancements drive governments to innovate in public service delivery. One of the steps that can be taken is to develop innovations in the implementation of e-government. In Indonesia, the concept of e-government was first introduced in 2001[1], following the issuance of Presidential Instruction No. 6 of 2001 on Telematics (Telecommunications, Media, and Informatics) [2].

Digitization in the government sector enables the provision of more effective and transparent public services while also encouraging citizen engagement in decision-making processes. In today's digital era, governments continue to strive for efficiency optimization and service improvement through various technological innovations [3]. Based on the Road Traffic and Transportation Law No. 22 of 2009 [4], the Sistem One Roof Integrated Administration (One-Stop Administration System), also known as SAMSAT, is an administrative system used in Indonesia to manage non-tax state revenue and motor vehicle tax collection. Due to the rapid expansion of internet usage, the Indonesian government, through the Traffic Corps of the National Police (POLRI), introduced the SIGNAL application[5].

Motor vehicle payment services can be conducted online through a platform called Samsat Digital Nasional (SIGNAL). The Directorate General of Taxes of the Ministry of Finance of the Indonesian government developed this application to help citizens fulfill their tax obligations. The SIGNAL application has been downloaded by more than one million users and can be found on the Google Play Store.[6]. The main goal of Samsat is to provide efficient and integrated administrative services and ensure that vehicle owners fulfill their tax obligations. [3]. In this regard, the SIGNAL application represents a significant step toward modernizing the motor vehicle taxation system through e-government.

Currently, the SIGNAL application has received feedback from its users. There are approximately 172,000 user reviews on the Google Play Store, with many users expressing complaints about technical issues, data errors, and dissatisfaction with the application's performance. A previous study utilized the Naïve Bayes algorithm on 2,000 reviews [3], but an improper implementation of the algorithm resulted in less accurate sentiment analysis of user reviews for the SIGNAL application. Moreover, traditional machine learning methods such as Naïve Bayes are less effective in understanding complex language patterns and sentence context, which play a crucial role in sentiment analysis. Additionally, data imbalance in user reviews can also degrade the performance of classification models, as the model tends to be biased toward the majority class. These concerns highlight the importance of further evaluation of the application's reliability.

Manuscript received February 8, 2025. This work was supported in part by Informatics Engineering Department of Maulana Malik Ibrahim Islamic State University.

Ratna Savitri is with the Faculty of Engineering, Informatics Study Program, Bina Insan University, Lubuklinggau, South Sumatra, Indonesia. (corresponding author provide phone +6285758169739; email: ratnasavitri6@gmail.com)

Fido Rizki is with the Faculty of Engineering, Informatics Study Program, Bina Insan University, Lubuklinggau, South Sumatra, Indonesia.

Ahmad Sobri is with the Faculty of Engineering, Informatics Study Program, Bina Insan University, Lubuklinggau, South Sumatra, Indonesia.

Several studies have been conducted to identify effective methods for sentiment analysis of user reviews in applications. The first study focused on analyzing user reviews of the SIGNAL application on Google Play Store using the Naïve Bayes method on 2,000 review data. The results showed an accuracy of 63.61%, a precision of 92.19%, and a recall of 61.52% [3]. The second study explored sentiment analysis on the DANA application using the Bidirectional Encoder Representations from Transformers (BERT) method on 13.231 review data. By implementing IndoBERT, the study achieved an accuracy of 98% and a validation accuracy of 93% after training for 10 epochs with a 70:30 data split ratio [7]. Another study analyzed user reviews of the Genshin Impact game application using BERT, utilizing 12,000 datasets from the Google Play Store. The results showed the highest precision value of 0.86%, recall of 0.78%, and an F1-score of 0.82%, indicating that the model effectively captured sentiment [8].

This study utilizes IndoBERT, a variant of BERT (Bidirectional Encoder Representations from Transformers), for sentiment analysis, optimized using the Random Over-Sampling technique to address data imbalance. BERT is a Deep Learning model developed by Google and released in 2018. In the natural language processing (NLP) translation process, this model is used to represent words in context. This enables a better understanding of word meanings based on the context of the sentence. [9]. IndoBERT, on the other hand, is a BERT model specifically designed for the Indonesian language and has proven to be effective in NLP tasks, including sentiment analysis, due to its ability to understand context and meaning in complex Indonesian sentences [8]. Compared to traditional approaches, IndoBERT has shown superior performance in sentiment classification tasks, particularly in handling nuanced language structures and imbalanced datasets. This study aims to identify a more accurate sentiment analysis method compared to previous approaches and to gain better insights into user perceptions of the SIGNAL application. By providing a more reliable sentiment analysis model, this research contributes to enhancing the evaluation process of government digital services and improving user experience in e-government applications

II. Literature Review

A. Sentiment analysis

In natural language processing (NLP), sentiment analysis, also known as opinion mining, is a technique aimed at identifying and conveying emotions, opinions, evaluations, attitudes, subjectivity, and judgments present in a given text. [7]. In the context of e-government, sentiment analysis is used to understand public satisfaction with digital services. By analyzing user reviews, the government can improve the quality of application-based services such as SIGNAL.

B. User Reviews

One of the features provided by the Google Play Store is user reviews, which allow users to give feedback or reviews through ratings and comments for the applications they download [10]. In sentiment analysis, user reviews serve as the primary data source that can be analyzed to identify public opinion trends toward a digital service.

C. National Digital Samsat (SIGNAL)

SIGNAL is a platform that provides vehicle tax payment services via the internet. The Indonesian government, through the Directorate General of Taxes of the Ministry of Finance, developed this application to facilitate tax payments for the public. SIGNAL can be downloaded from the Google Play Store and has been used by more than one million users [6].

D. Machine Learning

The field of artificial intelligence known as machine learning focuses on developing algorithms and techniques that enable computer systems to learn from data and improve their performance over time without explicit programming [11].

E. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a deep learning model that has achieved advanced performance in various Natural Language Processing (NLP) tasks [12]. This model can understand the meaning of words based on their context within a sentence, unlike traditional approaches that only consider words individually.

F. IndoBERT

IndoBERT is a BERT model specifically designed for the Indonesian language. This model has proven effective in natural language processing tasks such as sentiment analysis because it can understand the context and meaning of complex Indonesian sentences. [9]. Compared to other Transformer models such as RoBERTa and XLNet, IndoBERT is superior in Indonesian NLP tasks because it has been trained on a large Indonesian language corpus.

G. Random Over Sampling (ROS)

Random Over Sampling is a technique that involves randomly adding data from the minority class into the training dataset. This process is repeated until the number of data points in the minority class is equal to that of the majority class [13].

III. Research mothodology

A. Research Method

This study uses an exploratory research method. An exploratory research method is conducted to search for new ideas or relationships from specific phenomena [14]. This study employs sentiment analysis using IndoBERT, a BERT-based model, optimized with Random Over Sampling (ROS) to address data imbalance. This study employs sentiment analysis using IndoBERT, a BERT-based model, optimized with Random Over Sampling (ROS) to address data imbalance. There is a conceptual flowchart of the research that shows the stages from data collection to model evaluation, as depicted in Figure 1.

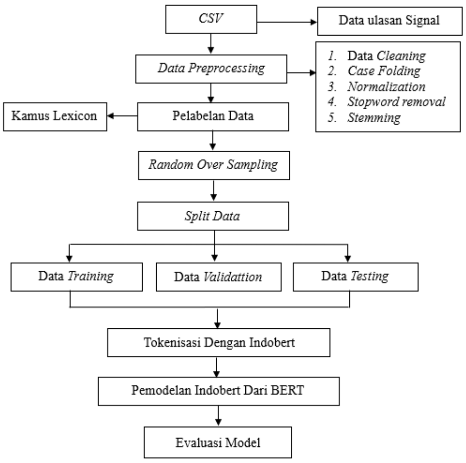


Figure 1 Research Concept Diagram

B. *Data Collection Techniques*

1) Primary Data

Primary data refers to data collected directly by the researcher [14]. In this study, the primary data used are user reviews of the Samsat Digital Nasional (SIGNAL) application, available on the Google Play Store, with 20,000 reviews in Indonesian.

2) Secondary Data

Secondary data refers to data that has been collected by others previously. This type of data is gathered by parties not directly related to the research study but for other purposes and at a different time in the past [14].

C. *Analysis Method*

The researcher employs the sentiment analysis method using IndoBERT from BERT (BiDirectional Encoder Representations from Transformers). IndoBERT was chosen because it can understand the context and meaning of complex Indonesian sentences and has proven to be effective in natural language processing tasks such as sentiment analysis.
[9].

D. *Testing and Data Processing Method*

1) Testing Method

This testing method aims to evaluate the performance of the classification method. The results from the confusion matrix table are used to measure classification performance by calculating the values for accuracy, precision, recall, and F1-Score, as follows:

a) Accuracy

Accuracy is an assessment conducted to determine the overall classification capability of a model.

$$Accuracy = \frac{TP + TN}{(TP+TN+FP+FN)} \text{ x } 100\% \quad (1)$$

b) Precision

Precision is the percentage of accuracy between the information requested by the user and the response provided by the system. The formula to determine Precision is as follows:

$$Precision = \frac{TP}{(TP+FP)} \text{ x } 100\% \quad (2)$$

c) Recall

Recall is an evaluation metric that describes how well a model correctly identifies the positive class. The formula to determine Recall is as follows:

$$Recall = \frac{TP}{(TP+FN)} \text{ x } 100\% \quad (3)$$

d) F1-Score

F1 Score is an evaluation metric that shows the balance between Recall and Precision. The formula to determine the F1 Score is as follows:

$$F1\ Score = \frac{(2 \text{ x } Recall \text{ x } Precision)}{Recall+Precision} \text{ x } 100 \quad (4)$$

2) Data Processing

In this study, the processing of user reviews is carried out through the following stages:

a) Preprocessing Stage

Before processing the data, the first step taken is preprocessing using the Python programming language. This stage includes several processes, as shown in Figure 2.

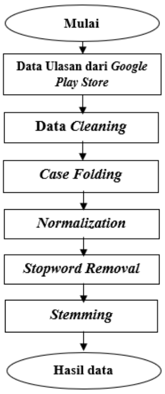


Figure 2 Preprocessing Stage

To ensure data quality, several data cleaning stages are performed before sentiment analysis is conducted:

1) Data Cleaning

At this stage, phrases in the dataset will be cleaned of duplicate data, excessive spaces, symbols, and emojis. This process is carried out to prevent negative impacts on the analysis results, which could lead to suboptimal classification [9].

2) Case Folding

At the case folding stage, uppercase letters in the data are converted to lowercase. This process facilitates text search and processing, as not all text uses uppercase letters consistently.

3) Normalization

Normalization is the process of converting text into a standard or normal form, making analysis easier to perform.

4) Stopword Removal

Words that are not highly significant in text analysis are removed through a process called Stopwords Removal. These words typically do not contribute to understanding or providing context to a sentence.[15].

5) Stemming

Stemming is the process of converting each word to its root form according to the rules found in the Indonesian Dictionary (KBBI) [10].

b) Labeling Stage

Data labeling is used to determine whether a phrase falls into the Positive, Neutral, or Negative analysis category. [10]. In this study, the labeling process is conducted using a Lexicon Dictionary.

c) Data Visualization

Data visualization is a method for presenting information and data in graphic form, making it easier to understand and explore the information. The main goal of data visualization is to help users comprehend the structure and meaning of the data, as well as support decision-making by identifying hidden patterns [3].

d) Data Splitting

The next stage is data splitting. The dataset was split in a 70:15:15 ratio for training, validation, and testing [9].

e) Random Over Sampling

ROS refers to the process of randomly adding minority class data to the training set until the number of minority class samples equals that of the majority class [9].

f) Tokenization with IndoBERT

Ttokenization is the process of converting sensitive data into non-sensitive data. This stage includes the processes of token embeddings, segment embeddings, and position embeddings. The token "[CLS]" is used to represent the start of a sequence, whiile "[SEP]" is used to separate segments (text)

g) IndoBERT Model Classification from Bidirectional Encoder Representations from Transformers (BERT)

IndoBERT is a pre-trained BERT-based model developedd for the Indonesian language [9]. The flow of applying IndoBERT can be seen in Figure 3 below:

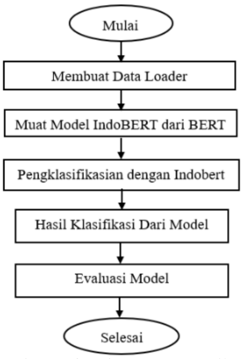

Mulai
Membuat Data Loader
Muat Model IndoBERT dari BERT
Pengklasifikasian dengan Indobert
Hasil Klasifikasi Dari Model
Evaluasi Model
Selesai

Figure 3 Flow of IndoBERT Application

IV. Results and Discussion

In this study, the implementation was carried out using Python through the Jupyter platform and Google Colab. To run the Python program, the use of libraries is required. A library is a collection of reusable code that can be used in various programs. To achieve the desired results in this research, several stages were carried out, including data collection through review scraping, data preprocessing, labeling process, classification, and evaluation.

A. Data Collection

In this study, the author used 20,000 user review data. The data was collected from December 2023 to October 2024. After that, the collected data will be downloaded and saved in CSV (Comma Delimited) file format for the next stages.

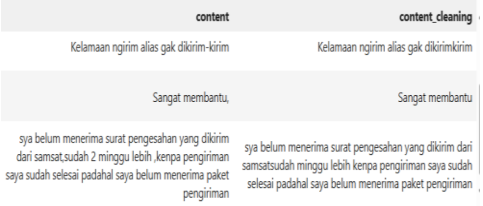
B. Data Preparation Stage

1) *Preprocessing Stage*

The next stage is Preprocessing, which is the step to prepare the data for use in the following processes.

a. Data Cleaning

In this stage, sentences in the dataset will be cleaned from duplicate data, excess spaces, symbols, and emoticons to prevent negative impacts on the analysis results, which could cause suboptimal classification. The result of the Data Cleaning can be seen in Figure 4.

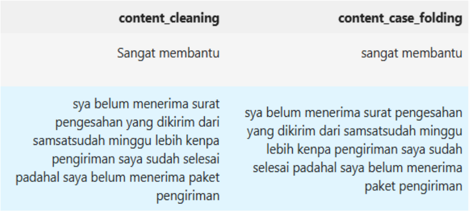


content	content_cleaning
Kelamaan ngirim alias gak dikirim-kirim	Kelamaan ngirim alias gak dikirimkirim
Sangat membantu,	Sangat membantu
sya belum menerima surat pengesahan yang dikirim dari samsat,sudah 2 minggu lebih ,kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman	sya belum menerima surat pengesahan yang dikirim dari samsatsudah minggu lebih kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman

Figure 4 Data Cleaning Results

b. Case Folding

In the Case Folding stage, the review data is processed by converting all uppercase letters to lowercase. The result of the Case Folding stage can be seen in Figure 5.

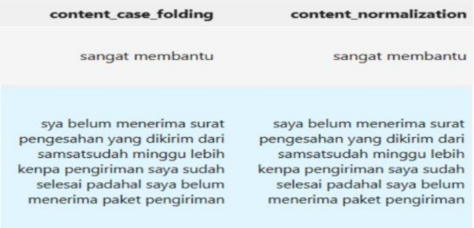


content_cleaning	content_case_folding
Sangat membantu	sangat membantu
sya belum menerima surat pengesahan yang dikirim dari samsatsudah minggu lebih kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman	sya belum menerima surat pengesahan yang dikirim dari samsatsudah minggu lebih kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman

Figure 5 Case Folding Results

c. Normalization

In the Normalization stage, the text is converted to a standard or normalized form using a "kamuskatabaku" (standard word dictionary) obtained from GitHub. This dictionary contains a list of non-standard words along with their corresponding standard words. The result of the Normalization stage can be seen in Figure 6

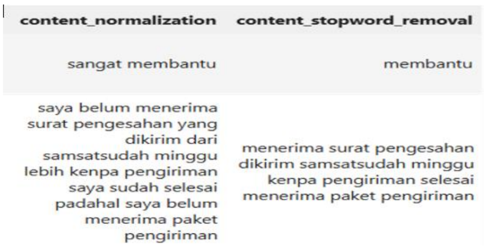


content_case_folding	content_normalization
sangat membantu	sangat membantu
sya belum menerima surat pengesahan yang dikirim dari samsatsudah minggu lebih kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman	saya belum menerima surat pengesahan yang dikirim dari samsatsudah minggu lebih kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman

Figure 6 Normalization Results

d. Stopword Removal

In the Stopword Removal stage, words that do not have significant meaning in the text analysis will be removed. These words typically do not contribute to the understanding or context of the sentence. The result of the Stopword Removal stage can be seen in Figure 7.

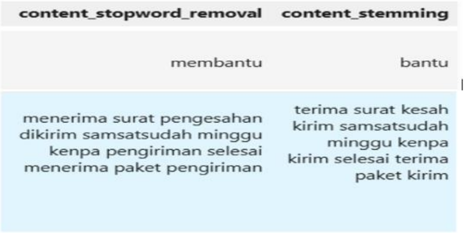


content_normalization	content_stopword_removal
sangat membantu	membantu
saya belum menerima surat pengesahan yang dikirim dari samsatsudah minggu lebih kenpa pengiriman saya sudah selesai padahal saya belum menerima paket pengiriman	menerima surat pengesahan dikirim samsatsudah minggu kenpa pengiriman selesai menerima paket pengiriman

Figure 7 Stopword Removal Results

e. Stemming

In the Stemming stage, each word will be transformed into its root form based on the rules from the Kamus Besar Bahasa Indonesia (KBBI). The result of the Stemming stage can be seen in Figure 8.



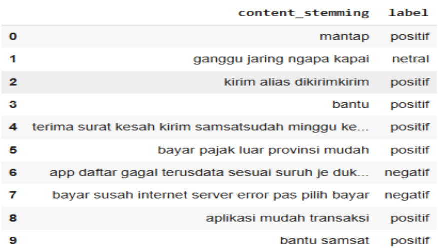
content_stopword_removal	content_stemming
membantu	bantu
menerima surat pengesahan dikirim samsatsudah minggu kenpa pengiriman selesai menerima paket pengiriman	terima surat kesah kirim samsatsudah minggu kenpa kirim selesai terima paket kirim

Figure 8 Stemming Results

C. Data Processing Stage

1) *Labelling*

The user review data that has been collected and processed through the Preprocessing stage will then be labeled as positive, neutral, or negative using a lexicon dictionary obtained from GitHub. The result of this labelling stage is shown in Figure 9.

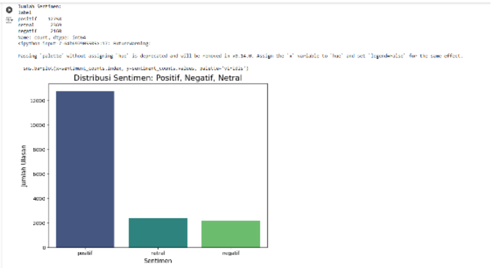


	content_stemming	label
0	mantap	positif
1	ganggu jaring ngapa kapai	netral
2	kirim alias dikirimkirim	positif
3	bantu	positif
4	terima surat kesah kirim samsatsudah minggu ke...	positif
5	bayar pajak luar provinsi mudah	positif
6	app daftar gagal terusdata sesuai suruh je duk...	negatif
7	bayar susah internet server error pas pilih bayar	negatif
8	aplikasi mudah transaksi	positif
9	bantu samsat	positif

Figure 9 Labelling Results

2) *Data Visualization*

The next step is to visualize the data to make it easier to understand and more appealing. In this stage, the Matplotlib library is used to present the data in the form of a chart (bar diagram). This visualization aims to display the number of data points labeled as positive, neutral, and negative. The result of this visualization is shown in Figure 10.



```
Jumlah Sentimen:
label
positif    12758
netral      2369
negatif     2160
Name: count, dtype: int64
<ipython-input-7-[illegible]>:17: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values, palette='viridis')
```

Figure 10 Visualization of Labelling Results

The figure showing the data visualization in the form of a bar chart indicates that the highest number of reviews is found under the positive label, with a total of 12,758 reviews. Meanwhile, the neutral label has 2,160 reviews, and the negative label has 2,369 reviews.

The visualization of reviews for the National Digital Samsat application is also presented in the form of a Word Cloud. A Word Cloud is a graphic representation that displays words from a text, where the most frequently occurring words are shown in larger and more prominent sizes. The result of the Word Cloud visualization can be seen in Figure 11.

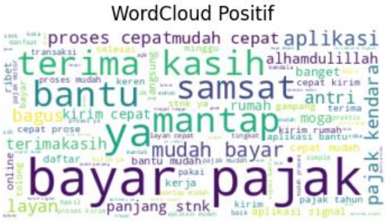


Figure 11 Word Cloud of Positive Reviews

The positive Word Cloud highlights frequently used terms in positive reviews, including “pay”, “tax”, “help”, “thank you”, and “great”, which reflect the ease of payment, assistance, and service satisfaction.

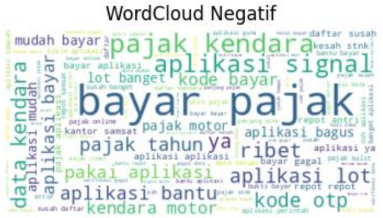


Figure 12 Word Cloud of Negative Reviews

The negative WordCloud image displays words that frequently appear in negative reviews of National Digital Samsat users, such as "pay", "tax", "vehicle", "code", and "complicated". These words reflect user issues, such as difficulties in payment, OTP code problems, and app complexity, which need to be addressed to improve the user experience.

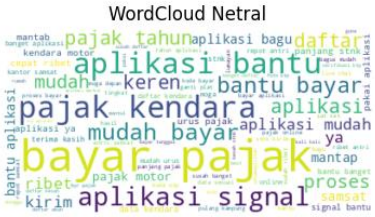
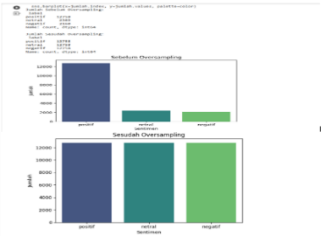


Figure 13 Word Cloud of Neutral Reviews

The neutral WordCloud image displays words that frequently appear in neutral reviews of National Digital Samsat users, such as "pay", "tax", "app", "vehicle", and "help". These reviews tend to be informative or descriptive, without expressing a clear sentiment, either positive or negative.

3) Random Over Sampling

The next step is to apply Random Over Sampling because the labeled data shows a class imbalance (Data Imbalance). This process aims to balance the number of data in the minority class with the majority class. The data before and after Random Over Sampling is applied is displayed in a graph, as shown in Figure 14.



```
sns.barplot(x=jumlah.index, y=jumlah.values, palette=color)
Jumlah Sebelum Oversampling:
 label
positif    12758
netral      2360
negatif     2160
Name: count, dtype: int64

Jumlah Sesudah Oversampling:
 label
positif    12758
netral     12758
negatif    12758
Name: count, dtype: int64
```

Figure 14 Visualization of data before and after ROS

The dataset initially contained 17.287 entries: 12.758 positive, 2.360 neutral, and 2.160 negative labels. After applying Random Over Sampling, the total data increases to 38,274, with each sentiment class having 12,758 entries.

4) Data Split Stage

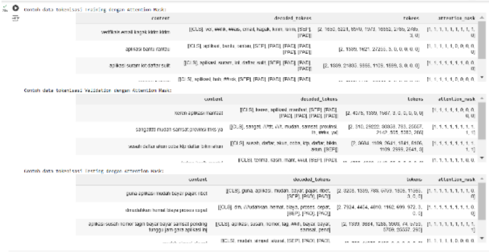
The next step is to split the data, which involves dividing the data into three parts: training data, validation data, and test data. In this study, a 70:15:15 proportion is used for the data split [9], as shown in Table 1.

Table 1 Data Split Results

Rasio	Hasil Split Data		
	Training	Validasi	Testing
70:15:15	26791	5742	5741

5) Tokenization with IndoBERT

he next step is tokenization with the IndoBERT model. This process converts each sentence into numbers, which are then encoded before being input into the model. Tokenization is carried out using the BERT library, specifically the BERT-Tokenizer. The results of tokenization can be seen in Figure 15.



```
Contoh data tokenisasi Training dengan Attention Mask:
                        content                        decoded_tokens                                 tokens                  attention_mask
verifikasi email kapok kirim kirim    [[CLS], ver, ##ifik, ##asi, email, kapok, kirim, kirim, [SEP], [PAD]]   [2, [illegible] 3, 0]   [1, 1, 1, 1, 1, 1, 1, 1, 1, 0]
aplikasi bantu rantau    [[CLS], aplikasi, bantu, rantau, [SEP], [PAD], [PAD], [PAD], [PAD], [PAD]]   [2, 1339, 1621, 27255, 3, 0, 0, 0, 0, 0]   [1, 1, 1, 1, 1, 0, 0, 0, 0, 0]
aplikasi suram lot daftar sulit    [[CLS], aplikasi, suram, lot, daftar, sulit, [SEP], [PAD], [PAD], [PAD]]   [2, 1339, 21835, 9866, 1109, 1699, 3, 0, 0, 0]   [1, 1, 1, 1, 1, 1, 1, 0, 0, 0]
[illegible]    [[CLS], aplikasi, bnh, ##risk, [SEP], [PAD], [PAD], [PAD], [PAD]   [illegible]   [1, 1, 1, 1, 1, 0, 0, 0

Contoh data tokenisasi Validation dengan Attention Mask:
                        content                        decoded_tokens                                 tokens                  attention_mask
keren aplikasi manfaat    [[CLS], keren, aplikasi, manfaat, [SEP], [PAD], [PAD], [PAD], [PAD], [PAD]]   [2, 4575, 1339, 1667, 3, 0, 0, 0, 0, 0]   [1, 1, 1, 1, 1, 0, 0, 0, 0, 0]
sangattt mudah samsat provinsi thks ya    [[CLS], sangat, ##tt, ##t, mudah, samsat, provinsi, th, ##ks, ya]   [2, 310, 29222, 30858, 783, 25557, 2142, 505, 5383, 260]   [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
susah daftar akun coba ktp daftar bikin akun    [[CLS], susah, daftar, akun, coba, ktp, daftar, bikin, akun, [SEP]]   [2, 3604, 1109, 2641, 1841, 6106, 1109, 2999, 2641, 3]   [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[illegible]    [[CLS], terima, kasih, mant, ##ul, [SEP], [PAD]   [illegible]   [1, 1, 1, 1, 1, 1, 0, 0

Contoh data tokenisasi Testing dengan Attention Mask:
                        content                        decoded_tokens                                 tokens                  attention_mask
guna aplikasi mudah bayar pajak ribet    [[CLS], guna, aplikasi, mudah, bayar, pajak, ribet, [SEP], [PAD], [PAD]]   [2, 3208, 1339, 783, 6709, 1606, 11566, 3, 0, 0]   [1, 1, 1, 1, 1, 1, 1, 1, 0, 0]
dimudahkan hemat biaya proses cepat    [[CLS], dim, ##udahkan, hemat, biaya, proses, cepat, [SEP], [PAD], [PAD]]   [2, 7924, 4464, 4810, 1162, 699, 972, 3, 0, 0]   [1, 1, 1, 1, 1, 1, 1, 1, 0, 0]
aplikasi susah nomor tagih bayar bayar samsat pending tunggu jam gara aplikasi inj    [[CLS], aplikasi, susah, nomor, tag, ##ih, bayar, bayar, samsat, pend]   [2, 1339, 3604, 1288, 3008, 74, 6709, 6709, 25557, 290]   [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[illegible]    [[CLS], mudah, singel, akurat, [SEP], [PAD], [PAD],   [illegible]   [1, 1, 1, 1, 1, 0, 0, 0
```

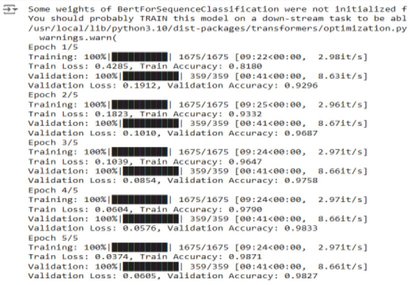
Figure 15 Tokenization Results with IndoBERT

6) Classification with IndoBERT Model from BERT (Bidirectional Encoder Representations from Transformers)

Training is performed on the training and validation data that have been split previously with a 70:15:15 ratio [10]. In this study, the classification process is carried out for 5 epochs with a batch size of 16, a learning rate of 2e-5, and the Adam optimizer. The results of the classification can be seen in Table 2 and Figure 4.15 below:

Table 2 IndoBERT Classification Results

Epoch	*Train*		*Validation*	
	Loss	*Accuracy*	*Loss*	*Accuracy*
1	0.4285	0.8180	0.1912	0.9296
2	0.1823	0.9332	0.1010	0.9687
3	0.1039	0.9647	0.0854	0.9758
4	0.0604	0.9790	0.0576	0.9833
5	0.0374	0.9871	0.0605	0.9827



```
Some weights of BertForSequenceClassification were not initialized f
You should probably TRAIN this model on a down-stream task to be abl
/usr/local/lib/python3.10/dist-packages/transformers/optimization.py
  warnings.warn(
Epoch 1/5
Training: 100%|██████████| 1675/1675 [09:22<00:00,  2.98it/s]
Train Loss: 0.4285, Train Accuracy: 0.8180
Validation: 100%|██████████| 359/359 [00:41<00:00,  8.63it/s]
Validation Loss: 0.1912, Validation Accuracy: 0.9296
Epoch 2/5
Training: 100%|██████████| 1675/1675 [09:25<00:00,  2.96it/s]
Train Loss: 0.1823, Train Accuracy: 0.9332
Validation: 100%|██████████| 359/359 [00:41<00:00,  8.67it/s]
Validation Loss: 0.1010, Validation Accuracy: 0.9687
Epoch 3/5
Training: 100%|██████████| 1675/1675 [09:24<00:00,  2.97it/s]
Train Loss: 0.1039, Train Accuracy: 0.9647
Validation: 100%|██████████| 359/359 [00:41<00:00,  8.66it/s]
Validation Loss: 0.0854, Validation Accuracy: 0.9758
Epoch 4/5
Training: 100%|██████████| 1675/1675 [09:24<00:00,  2.97it/s]
Train Loss: 0.0604, Train Accuracy: 0.9790
Validation: 100%|██████████| 359/359 [00:41<00:00,  8.66it/s]
Validation Loss: 0.0576, Validation Accuracy: 0.9833
Epoch 5/5
Training: 100%|██████████| 1675/1675 [09:24<00:00,  2.97it/s]
Train Loss: 0.0374, Train Accuracy: 0.9871
Validation: 100%|██████████| 359/359 [00:41<00:00,  8.66it/s]
Validation Loss: 0.0605, Validation Accuracy: 0.9827
```

Figure 16 IndoBERT Classification Results

From the table and figure, we can observe that the IndoBERT model training was conducted for 5 epochs, with a significant improvement in performance. Initially, the Train Loss was relatively high (0.4285) with a Train Accuracy of 81.80%, but it continued to improve, reaching a Train Loss of 0.0374 and Train Accuracy of 98.71% in the final epoch. The Validation Loss and Validation Accuracy also showed consistent performance, with Validation Accuracy peaking at 98.33% in the 4th epoch. This trend indicates that the model successfully learned the patterns well. To support this understanding, a visualization in the form of a graph clearly shows the trend of performance improvement during training. The visualization can be seen in Figure 16.

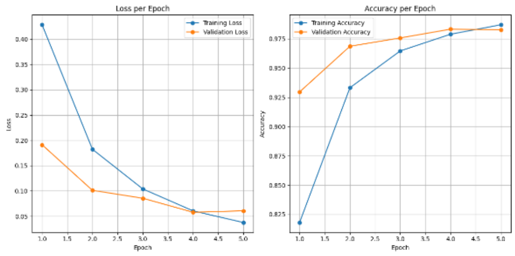
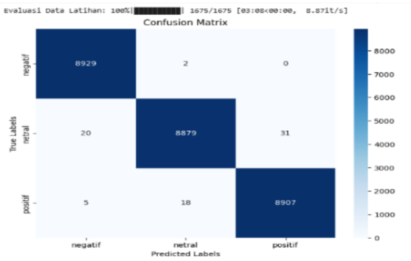


Figure 17 Visualization of training loss and accuracy results

7) Model Evaluation.

After the training process, the model is evaluated using both training and testing data. Below are the evaluation results using the training data:



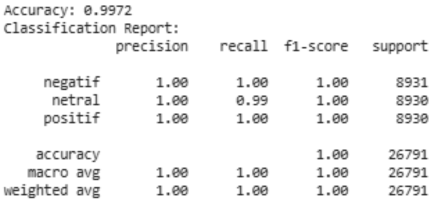
```
Evaluasi Data Latihan: 100%|██████████| 1675/1675 [03:08<00:00,  8.87it/s]
```

Figure 18 Training Data Confusion Matrix

The model evaluation results on the training data achieved an accuracy of 99.72%, indicating an excellent ability to recognize training data patterns. The misclassification rate is very low, with a minimal number of false positives (FP) and false negatives (FN). The classification details are as follows:

1) Negative class: 8929 samples were correctly classified, while 20 samples were misclassified as neutral.
2) Neutral class: 8879 samples were correctly classified, while 31 samples were misclassified as positive.
3) Positive class: 8,907 samples were correctly classified, while 5 samples were misclassified as negative.

After that, to gain a deeper understanding of the model's performance, a classification report will be calculated. The classification report results for the training data can be seen in Figure 19.



```
Accuracy: 0.9972
Classification Report:
              precision    recall  f1-score   support

     negatif       1.00      1.00      1.00      8931
      netral       1.00      0.99      1.00      8930
     positif       1.00      1.00      1.00      8930

    accuracy                           1.00     26791
   macro avg       1.00      1.00      1.00     26791
weighted avg       1.00      1.00      1.00     26791
```

Figure 19 Training Data Classification Report

In the evaluation of the training data, the model showed a very high accuracy of 99.72%, with precision, recall, and F1-score metrics close to 1.00 for all classes (negative, neutral, and positive). This result indicates that the model has learned the patterns in the training data very well, with a very minimal error rate.

Afterward, the model was evaluated to measure how well it performs in predicting unseen data using the Testing data. The evaluation was conducted using a confusion matrix, which provides a clearer picture of the model's ability to classify data correctly. The results of the confusion matrix can be seen in the image.

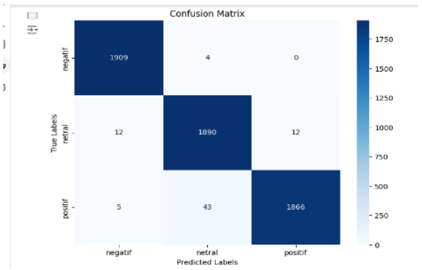
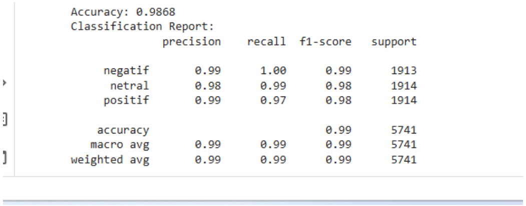


Figure 20 Testing Data Confusion Matrix

On the test data, the model still demonstrates high performance with an accuracy of 98.68%. However, there is an increase in misclassification compared to the training data. The classification details are as follows:

1) Negative class: 1909 samples were correctly classified, while 12 samples were misclassified as neutral.
2) Neutral class: 1890 samples were correctly classified, while 43 samples were misclassified as positive.
3) Positive class: 1866 samples were correctly classified, while 5 samples were misclassified as negative.

After that, to gain a deeper understanding of the model's performance, the classification report will be calculated. This classification report includes important metrics such as precision, recall, F1-score, and support for each class, allowing us to evaluate the model's performance in more detail. The results of the classification report can be seen in the image.



```
Accuracy: 0.9868
Classification Report:
              precision    recall  f1-score   support

     negatif       0.99      1.00      0.99      1913
      netral       0.98      0.99      0.98      1914
     positif       0.99      0.97      0.98      1914

    accuracy                           0.99      5741
   macro avg       0.99      0.99      0.99      5741
weighted avg       0.99      0.99      0.99      5741
```

Figure 21 Testing Data Classification Report

The classification report provides an assessment of the model's performance through the metrics of Precision, Recall, and F1-Score. For the negative review category, the model achieved a precision of 0.99, recall of 1.00, and an F1-Score of 0.99, indicating a very high accuracy in identifying negative reviews. For the neutral category, a precision of 0.98, recall of 0.99, and F1-Score of 0.98 suggest very good performance, although slightly below the negative category. Meanwhile, for the positive review category, precision was recorded at 0.99, recall at 0.97, and F1-Score at 0.98, indicating that the model faced slightly more difficulty in classifying positive reviews compared to the other two categories. After evaluation using the Testing data, overall, the model achieved a total accuracy of 98.68%, with very high average metrics across all categories.

V. Conclusion

From the research conducted on sentiment analysis of user reviews for the National Digital Vehicle Registration System (SIGNAL) application on Google Play Store using the IndoBERT model from BERT, the following conclusions can be drawn:

1. The data used consists of 20,000 reviews, obtained through scraping technique using the Google-Play-Scraper library on Google Colab. Afterward, the data was labeled using a Lexicon dictionary, resulting in 12,758 reviews labeled as positive, 2,160 reviews labeled as neutral, and 2,369 reviews labeled as negative. The labeled data shows a significant imbalance between the number of positive, negative, and neutral reviews. Therefore, Random Over Sampling technique was applied to balance the data distribution across all three categories.
2. The sentiment analysis performed using the BERT model from IndoBERT achieved an accuracy of 99% and a validation accuracy of 98% after training for 5 epochs with a data split of 70:30 (70% for training data, 15% for validation data, and 15% for test data). These results indicate that IndoBERT is highly effective in sentiment analysis of user reviews.
3. Based on the model's testing using the confusion matrix, overall, the model achieved an accuracy of 99.72% on the training data and 98.68% on the testing data.

VI. Suggestions

Based on the findings and limitations of this research, several suggestions can be made:

1. Dataset Enhancement: Use a larger and more diverse dataset to improve the model's accuracy, covering various sentiments and topics, so the model can better understand the context and language variations of users.
2. Alternative NLP Models: In addition to BERT, models such as XLNet or RoBERTa could be tested for a more comprehensive comparison in sentiment analysis.
3. Training Efficiency: Given the high computational demands, it is recommended to explore techniques like distillation or more resource-efficient fine-tuning.
4. Validity and Generalization: Evaluate with datasets from other sources to reduce bias and improve the model's ability to understand reviews from various platforms.

VII. References

[1] W. A. Rahmadhani *et al.*, "Pemanfaatan Website Sebagai Bentuk Digitalisasi Pelayanan Publik Untuk Mewujudkan Transparansi di Dinas Sosial Provinsi Sumatera Utara dan Pemanfaatan Digitalisasi Pada Pendidikan Islam," *Edukasi Islam. J. Pendidik. Islam*, vol. 11, no. 1, pp. 1167–1182, 2022, doi: 10.30868/ei.v11i01.2979.

[2] Republik Indonesia, *Instruksi Presiden No. 6 Tahun 2001 Tentang Pengembangan Dan Pendayagunaan Telematika Di Indonesia Presiden Republik Indonesia*. Indonesia, 2001, pp. 1–11. [Online]. Available: https://jdih.esdm.go.id/peraturan/INPRES NO 6 TH 2001.pdf

[3] N. Nurzaman, N. Suarna, and W. Prihartono, "Analisis Sentimen Ulasan Aplikasi Threads Di Google Playstore Menggunakan Algoritma Naïve Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 967–974, 2024, doi: 10.36040/jati.v8i1.8708.

[4] R. Indoensia, "Undang Undang Republik Indonesia No. 22 Tahun 2009 Tentang Lalu Lintas dan Angkatan Umum," vol. 19, p. 19, 2009.

[5] S. Kacung, C. Pamungkas Putra Bagyana, and D. Cahyono, "Analisis Sentimen Terhadap Layanan Samsat Digital Nasional (Signal) Menggunakan Metode Svm," *J. Mnemon.*, vol. 7, no. 1, pp. 118–122, 2024, doi: 10.36040/mnemonic.v7i1.9557.

[6] I. F. Rahman, A. N. Hasanah, and N. Heryana, "Analisis Sentimen Ulasan Pengguna Aplikasi Samsat Digiital Nasional (Signal) Dengan Menggunakan Metode Naïve Bayes Classifier," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 963–969, 2024, doi:

10.23960/jitet.v12i2.4073.

[7] F. I. Septian, H. Ivana Lucia Kharisma, and Kamdan, "Implementasi Metode Bidirectional Encoder Representations from Transformers (BERT) untuk Analisis Sentimen Komentar Pengguna," vol. 3, no. 1, 2023.

[8] R. Kusnadi, Y. Yusuf, A. Andriantony, R. Ardian Yaputra, and M. Caintan, "Analisis Sentimen Terhadap Game Genshin Impact Menggunakan Bert," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 6, no. 2, pp. 122–129, 2021, doi: 10.36341/rabit.v6i2.1765.

[9] K. Cindy Pradhisa and R. Fajriyah, "Analisis Sentimen Ulasan Pengguna E-commerce di Google Play Store Menggunakan Metode IndoBERT," *Technol. Sci.*, vol. 6, no. 1, pp. 92–104, 2024, doi: 10.47065/bits.v6i1.5247.

[10] R. Kurniawan, H. O. L. Wijaya, and R. P. Aprisusanti, "Sentiment Analysis of Google Play Store User Reviews on Digital Population Identity App Using K-Nearest Neighbors," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 13, no. 2, pp. 170–178, 2024, doi: 10.32736/sisfokom.v13i2.2071.

[11] Ardi Mursyidi, "Penerapan Bidirectional Encoder Representations From Transformers (Bert) Untuk Analisis Sentimen Vaksin Covid-19 Pada Twitter," *UIN Suska Riau*, 2023.

[12] R. Mas, R. W. Panca, K. Atmaja1, and W. Yustanti2, "Analisis Sentimen Customer Review Aplikasi Ruang Guru dengan Metode BERT (Bidirectional Encoder Representations from Transformers)," *Jeisbi*, vol. 02, no. 3, pp. 55–62, 2021.

[13] M. Haris, A. Suharso, E. H. Nurkifli, P. S. Informatika, U. S. Karawang, and T. Timur, "ANALISIS SENTIMEN PADA GAME EFOOTBALL DI GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA INDOBERT," vol. 8, no. 6, pp. 12108–12121, 2024.

[14] Ms. Elvis F. Purba, SE and P. . Parulian Simanjuntak, MA, *E-Book Metodologi Penelitian*, vol. 11, no. 1. 2011, 2011. [Online]. Available: http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI

[15] B. Kurniawan, A. Ari Aldino, and A. Rahman Isnain, "Sentimen Analisis terhadap Kebijakan Penyelenggara Sistem Elektronik (PSE) Menggunakan Algoritma Bidirectional Encoder Representations from Transformers (Bert)," *J. Teknol. dan Sist. Inf.*, vol. 3, no. 4, pp. 98–106, 2022, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI