JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 8, No. 2, December 2025, pp. 1216~1231
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 152/E/KPT/2023
DOI: https://doi.org/10.36378/jtos.v8i2.4822

Utilization of IndoBERT Representation and Random Forest for Sentiment Analysis on User Reviews of Halodoc Pharmacy Services in Google Play

Hendry Fonda[1], Herianto[2], Yuda Irawan[3]
Faculty of Computer Science, Hang Tuah University Pekanbaru, Indonesia

Article Info

Article history:

Received 12 01, 2025
Revised 12 16, 2025
Accepted 12 24, 2025

Keywords:

Sentiment Analysis,
IndoBERT,
Random Forest,
Google Play User Reviews.

ABSTRACT

With the growing use of digital healthcare platforms such as Halodoc, maintaining consistent service quality that meets user expectations is essential. User reviews on platforms like Google Play provide valuable insights into user perceptions. This study aims to classify user sentiments toward Halodoc's pharmacy services based on reviews obtained through web scraping from the Google Play Store. The analysis employs the pre-trained IndoBERT model to extract textual features, followed by sentiment classification using the Random Forest algorithm. This combination was selected for its efficiency with limited hardware resources and small dataset size. To enhance data diversity and minimize overfitting, simple augmentation methods such as random word deletion and synonym substitution were implemented. The expected outcomes include an effective sentiment classification model and visualizations of sentiment distributions (positive, negative, neutral). Furthermore, the study contributes to the development of sentiment analysis techniques for Indonesian-language data through an efficient and contextually relevant approach. The research outputs target publication in a nationally accredited (Sinta 4) journal and Intellectual Property Rights (IPR) registration. Ultimately, this study is expected to support the improvement of technology-based pharmacy services through the strategic application of machine learning.

Corresponding Author:

Hendry Fonda
Faculty of Computer Science,
Hang Tuah University
Pekanbaru, Indonesia
Email: ondahendry37@gmail.com

1. Introduction

In today's digital era, various applications have emerged to meet different aspects of human life, including desktop, tablet, and smartphone platforms (1). One example is pharmacy-based applications such as Halodoc, which are increasingly popular among Indonesian society due to their convenience in obtaining medicines and conducting health consultations efficiently. However, to ensure service quality and maintain user satisfaction, an in-depth analysis of user reviews on platforms such as the Google Play Store is necessary. Sentiment analysis of these reviews provides a direct reflection of user perceptions toward the services provided. In this context, the utilization of Natural Language Processing (NLP) technology becomes crucial to address challenges in processing natural language, such as spelling errors, writing style variations, and capturing subjective sentiment more accurately (1). Pre-trained models such as IndoBERT, which is specifically developed for the Indonesian language, can be used to extract feature representations from review texts (2), followed by the application of classification algorithms such as Random Forest to categorize user sentiments (3). Nevertheless, challenges arise when computational resources are limited and the dataset size is relatively small. To overcome this, simple data augmentation techniques, such as synonym replacement and random word deletion, can be applied to increase data diversity and reduce the risk of overfitting. Based on these issues, several research questions are formulated: (a) how to classify user sentiments toward Halodoc pharmacy services based on reviews collected from the Google Play Store; (b) how to utilize the IndoBERT pre-trained model as a feature extractor for sentiment classification; (c) how effective the Random Forest algorithm is in classifying sentiments under constraints of limited computational resources and small datasets; and (iv) how simple data augmentation techniques can enhance data diversity and mitigate overfitting risks. Machine learning, in this case, serves as the method to extract information and identify patterns from data, enabling predictions for future insights based on past data.

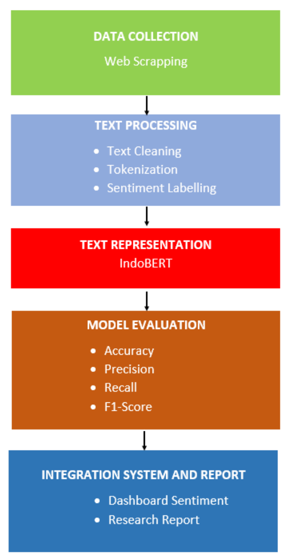

Figure 1. Research Flow Diagram

The research will be carried out in several structured stages. First, a literature review is conducted by the principal investigator to examine studies related to Natural Language Processing (NLP), text representation using IndoBERT, and sentiment classification with the Random Forest algorithm. The expected output of this stage is a comprehensive theoretical foundation and the identification of the best methods as well as research gaps to be addressed. The research stages can be described as follows Literature Review.

2. Research Method

This study employs a quantitative approach based on Natural Language Processing (NLP) and Machine Learning. The research will be conducted in several structured stages, starting from data collection, preprocessing, text representation, model training, system integration, and finally evaluation and reporting. The primary models used in this study are IndoBERT for text representation and Random Forest for sentiment classification:

1. Literature Review

   The principal investigator conducts a literature review related to Natural Language Processing (NLP), text representation using IndoBERT, and the Random Forest classification algorithm.

   a. Output: a comprehensive literature summary and theoretical foundation.

   b. Success Indicator: identification of the best methods and research gaps to be addressed.

2. Data Collection

   Member 1 is responsible for collecting user reviews of the Halodoc application from the Google Play Store using web scraping.

   a. Output: a raw dataset containing at least 2,000 reviews.

   b. Success Indicator: sufficient data collected for further processing.

   User reviews of the Halodoc application are collected from the Google Play Store using web scraping.

   Formally, the dataset is represented as:

$$D = \{(r_i, s_i) \mid i = 1, 2, \ldots, n\} \quad (1)$$

   where:

   r^i = review text

   s^i = rating score (1–5)

   n ≥ 2000 = number of reviews

3. Text Preprocessing

   The collected dataset is cleaned from special characters, tokenized, and labeled with sentiments based on user ratings.

   a. Output: a clean and structured dataset.

   b. Success Indicator: data ready for processing with the IndoBERT model.

   The dataset is cleaned (removing special characters, stopwords, emojis), tokenized, and labeled according to sentiment categories:

$$y_i = \begin{cases} \text{Negative}, & s_i \in \{1, 2\} \\ \text{Neutral}, & s_i = 3 \\ \text{Positive}, & s_i \in \{4, 5\} \end{cases} \quad (2)$$

   Thus, the processed dataset is:

$$D' = \{(x_i, y_i) \mid i = 1, 2, \ldots, n\} \quad (3)$$

Where x^i = cleaned review text, and y^i = sentiment label.

4. Text Representation

Member 2 converts the review texts into vector embeddings using IndoBERT, which captures the semantic meaning of the text.

a. Output: IndoBERT embeddings for the entire dataset.
b. Success Indicator: embeddings successfully generated and ready for model training.

Each review x^i is passed through **IndoBERT**, producing hidden states h^t for each token. A sentence embedding is obtained using mean pooling:

$$v_i = \frac{1}{T}\sum_{t=1}^{T} h_t \quad (4)$$

Where:

- T = number of tokens in review i
- $h_t \in \mathbb{R}^{768}$ = hidden vector of token t
- v_i = embedding representation of review i

The final representation set is:

$$X = \{v_1, v_2, \dots, v_n\}, \quad v_i \in \mathbb{R}^{768} \quad (5)$$

5. Model Training and Evaluation

The principal investigator, together with Member 2, trains the Random Forest model using IndoBERT embeddings and evaluates its performance with metrics such as accuracy, precision, recall, and F1-score.

a. Output: a trained sentiment classification model.
b. Success Indicator: model achieves accuracy above 80% and a high F1-score.

The embeddings X are used to train the **Random Forest classifier**. Each tree T^j makes a prediction, and the final output is determined by majority voting:

$$\hat{y}_i = \text{majority\_vote}\{T_j(v_i) \mid j = 1, 2, \dots, m\} \quad (6)$$

Where:

- m = number of decision trees
- $T_j(v_i)$ = prediction from tree j
- $\hat{y}_i$ = final predicted sentiment

The model is evaluated using standard metrics:

- **Accuracy**

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

- **Precision**

$$P = \frac{TP}{TP + FP}$$

- **Recall**

$$R = \frac{TP}{TP + FN}$$

- **F1-score**

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

6. Visualization and Interpretation
   Member 3 develops an analytical dashboard to display sentiment classification results in the form of trend graphs, sentiment distribution, and dominant keywords.
   a. Output: visualization of results in an interactive dashboard.
   b. Success Indicator: insights presented in an easy-to-understand format.
      Results are visualized in the form of sentiment distribution, keyword frequency, and trend analysis.
   The sentiment distribution function:

$$f(y) = \frac{\text{count}(y)}{n}, \quad y \in \{\text{Positive}, \text{Neutral}, \text{Negative}\} \tag{8}$$

7. Report Preparation
   The principal investigator prepares the final documentation of the entire research process and results.
   a. Output: the final research report and a draft scientific article.
   b. Success Indicator: report and draft publication ready for submission to a nationally accredited journal.

3. Results and Discussion

The initial stage of the research began with the preparation of the working environment to ensure that all required packages and dependencies were properly installed. This installation included core libraries such as pandas, scikit-learn, imbalanced-learn, transformers, torch, and matplotlib for analysis, modeling, and visualization purposes.

In addition, a random seed was set to guarantee that the experimental process could be reproduced with consistent results. Device checking (CPU or GPU) was also performed to optimize computational performance, particularly when running text representation using the transformer-based IndoBERT model.

With this setup step, the entire research process ranging from data preprocessing, model training, to system integration into the application could be executed within a stable and controlled environment, as illustrated in Figure 1 below.

```
# ====== LANGKAH 0: Setup & Install ======
import warnings, os, random, json, gc, re
warnings.filterwarnings("ignore")

# (Colab) Pastikan paket terpasang
%pip -q install -U pip
%pip -q install google-play-scraper pandas scikit-learn imbalanced-learn transformers torch matplotlib

import numpy as np, pandas as pd, matplotlib.pyplot as plt, torch, joblib
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, f1_score
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from imblearn.over_sampling import SMOTE
from transformers import AutoTokenizer, AutoModel

SEED = 42
random.seed(SEED); np.random.seed(SEED); torch.manual_seed(SEED)
if torch.cuda.is_available():
    torch.cuda.manual_seed_all(SEED)
device = "cuda" if torch.cuda.is_available() else "cpu"

print({
    "numpy": np.__version__,
    "pandas": pd.__version__,
    "torch": torch.__version__,
    "device": device
})
```

Figure 1. Required Package Installation

The first stage of the research was to collect user reviews of the Halodoc application from the Google Play Store for the period of January to March 2025. These reviews were used as the primary source for sentiment analysis, as they reflect user experiences, satisfaction, and complaints regarding the Halodoc digital pharmacy service.

The data collection process was carried out using the google-play-scraper library by utilizing the reviews() function, which provides a mechanism to scrape the latest reviews. To ensure that the data met the research requirements, a filter was applied based on the time range (January 1 – March 31, 2025). Each batch of scraped data was then further filtered to include only reviews within the specified period.

All collected reviews were stored in a Pandas DataFrame and exported into a CSV file (halodocx_reviews_jan_mar_2025.csv). This file served as the foundation for the subsequent stages, namely text preprocessing, representation with IndoBERT, and classification model training.

Through this step, the research obtained an actual and relevant dataset within the defined timeframe, allowing the analysis results to better reflect the state of Halodoc's service during the study period, as shown in Figure 2 below:

```python
# ====== LANGKAH 1: Scrape ulasan Halodoc (Jan–Mar 2025) ======
from google_play_scraper import reviews, Sort
from datetime import datetime

APP_ID = 'com.linkdokter.halodoc.android'
START = datetime(2025, 1, 1).date()
END   = datetime(2025, 3, 31).date()

def filter_by_date(results, start_date, end_date):
    return [r for r in results if start_date <= r['at'].date() <= end_date]

all_reviews, token = [], None
BATCH = 200

while True:
    batch, token = reviews(
        APP_ID, lang='id', country='id',
        sort=Sort.NEWEST, count=BATCH, continuation_token=token
    )
    if not batch:
        break

    # Keep hanya rentang tanggal
    filtered = filter_by_date(batch, START, END)
    all_reviews.extend(filtered)
    print(f"Terkumpul: {len(all_reviews)}")

    # Stop ketika batch sudah melewati awal periode
    if batch[-1]['at'].date() < START:
        break

# ke DataFrame & simpan
df_raw = pd.DataFrame(all_reviews)
csv_filename = 'halodocx_reviews_jan_mar_2025.csv'
df_raw.to_csv(csv_filename, index=False)
print("Disimpan:", csv_filename)
```

Figure 2. Scrapping Halodocs Reviews

The data in the table below represents the results of scraping user reviews of the Halodoc application from the Google Play Store during the period of January–March 2025. Each row corresponds to one review with several important recorded attributes. The data is presented in Figure 3 below:

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	e83917b4-27ec-4203-a2f3-f114bced83a4	Shanti Anjani	https://play-lh.googleusercontent.com/a/ACg8oc...	sangat membantu dikala dokter umum offline lib...	5	0	24.701	2025-03-31 22:50:46	Hai Shanti Anjani! Terima kasih sudah mengguna...	2025-04-01 00:03:06	24.701
1	ccb98d8e-5315-4a6e-b44a-d138a6db9534	langgeng septiana	https://play-lh.googleusercontent.com/a/ACg8oc...	bnr2 membantu	5	0	24.600	2025-03-31 20:18:23	Hai langgeng septiana! Terima kasih sudah meng...	2025-03-31 20:33:28	24.600
2	3399c2b8-6523-422e-bfc8-17b1fccf154a	Bayu Yudhistira	https://play-lh.googleusercontent.com/a/ACg8oc...	alhamdulillah ngebantu saat sakit gak usah bra...	5	0	NaN	2025-03-31 14:05:55	Hai Bayu Yudhistira! Terima kasih sudah menggu...	2025-03-31 20:32:02	NaN
3	bcfa3fd2-3a5d-4c88-9f91-ce7534ddd76f	Nurul Alya	https://play-lh.googleusercontent.com/a-/ALV-U...	sangat membantu	5	0	24.701	2025-03-31 13:27:56	Hai Nurul Alya! Terima kasih sudah menggunakan...	2025-03-31 20:09:53	24.701
4	bfcee29b-70f2-445b-b668-72506795f596	Cipto Prasetyo	https://play-lh.googleusercontent.com/a-/ALV-U...	sangat membantu	5	0	NaN	2025-03-31 10:33:07	Hai Cipto Prasetyo! Terima kasih sudah menggun...	2025-03-31 13:07:55	NaN
5	3a8b65ee-ca4c-43f6-ae42-8b2da7c2d675	L_ Posh Girl	https://play-lh.googleusercontent.com/a-/ALV-U...	fast respon meski lagi idul fitri	5	0	24.701	2025-03-31 04:34:16	Hai L_ Posh Girl! Terima kasih sudah menggunak...	2025-03-31 04:45:31	24.701
6	0319a0f6-32f8-4d6b-94e2-	USS_official	https://play-	konfirmasi segera	5	0	NaN	2025-03-30	Hai USS_official (rozyfatwari86)! Terima	2025-03-30	NaN

Figure 3. Dataset Halodoc Scrapping Result

Before further analysis was conducted, the scraped user review dataset from the Google Play Store needed to be normalized to ensure a consistent column structure. The first step was converting all column names into lowercase to prevent errors caused by capitalization differences, for example, Content and content. Next, since the dataset could contain different variations of text column names, a list of candidate names was prepared, including clean_text, review, content, text, ulasan, full_text, and comments. The system automatically searched for one of these candidates and set it as the main column with a standardized name, clean_text.

Once the main text column was determined, minimal cleaning was performed, such as converting values to strings, trimming leading and trailing spaces, and removing empty rows. As a result, a well-structured clean_text column was obtained and prepared for the IndoBERT representation stage. Some examples of entries in this column after normalization include: “sangat membantu dikala dokter umum offline lib…”, “bnr2 membantu”, and “alhamdulillah ngebantu saat sakit gak usah bra…”.

This preprocessing step is crucial as it ensures that the highly varied user-generated text is standardized and thus can be effectively processed in machine learning modeling. The results of this step are shown below:

```
# lowercase nama kolom
df = df.rename(columns={c: c.lower() for c in df.columns})

# deteksi kandidat kolom teks -> jadikan 'clean_text'
text_candidates = ["clean_text","review","content","text","ulasan","full_text","comments"]
col_text = None
for c in text_candidates:
    if c in df.columns:
        col_text = c
        break
assert col_text is not None, f"Kolom teks tidak ditemukan. Kandidat: {text_candidates}"

if col_text != "clean_text":
    df = df.rename(columns={col_text: "clean_text"})

# bersihkan minimal
df["clean_text"] = df["clean_text"].astype(str).str.strip()
df = df[df["clean_text"].str.len() > 0].reset_index(drop=True)
print("Kolom aktif:", df.columns.tolist())
display(df[["clean_text"]].head(3))
```

	clean_text
0	sangat membantu dikala dokter umum offline lib...
1	bnr2 membantu
2	alhamdulillah ngebantu saat sakit gak usah bra...

Figure 4. Lowercase column name

The preprocessing stage resulted in a primary column named clean_text, which contains normalized user reviews of the Halodoc application. Figure 5 presents a portion of the clean_text column, displaying entries from the first row to the last row. It can be observed that the user reviews are generally concise and straightforward, such as “sangat membantu dikala dokter umum offline lib…”, “bnr2 membantu”, and “Dokter ramah □ terimakasih”.

The total amount of data obtained after cleaning was 1,481 rows of reviews with one main text column. This stage ensured that all user review data had a standardized format, free from missing values, and ready to be used for the next processes, namely sentiment labeling based on rating scores and feature extraction using IndoBERT representation

	clean_text
0	sangat membantu dikala dokter umum offline lib...
1	bnr2 membantu
2	alhamdulillah ngebantu saat sakit gak usah bra...
3	sangat membantu
4	sangat membantu
...	...
1476	bad, udah beli salep harusnya dapet konsultasi...
1477	Dokter ramah 👍 terimakasih
1478	Sangat membantu
1479	Sangat membantu dan cepat
1480	Kerjasama dengan Astra, proses claim Reimburse...

1481 rows × 1 columns

Figure 5. Clean Text

The next stage after text normalization was sentiment labeling based on the rating scores provided by users in their Google Play Store reviews. This process was carried out using a simple rule: reviews with scores of 1–2 were labeled as Negative, a score of 3 was labeled as Neutral, and scores of 4–5 were labeled as Positive.

Thus, each review contained not only the cleaned text (clean_text) and its numerical rating (score), but also the corresponding sentiment label (sentiment). Figure 6 presents a portion of the labeling results, showing that reviews such as “sangat membantu dikala dokter umum offline lib…” with a score of 5 were labeled as Positive, while reviews like “bad, udah beli salep harusnya dapet konsultasi…” with a score of 1 were labeled as Negative.

The final dataset ready for modeling consisted of 1,481 rows with three main columns: clean_text, score, and sentiment. This stage ensured that the dataset was well-structured and prepared for further processing, namely feature representation using IndoBERT and classification using the Random Forest algorithm, as described in the following stages.

	clean_text	score	sentiment
0	sangat membantu dikala dokter umum offline lib...	5	positif
1	bnr2 membantu	5	positif
2	alhamdulillah ngebantu saat sakit gak usah bra...	5	positif
3	sangat membantu	5	positif
4	sangat membantu	5	positif
...	...	...	...
1476	bad, udah beli salep harusnya dapet konsultasi...	1	negatif
1477	Dokter ramah 👍 terimakasih	5	positif
1478	Sangat membantu	5	positif
1479	Sangat membantu dan cepat	5	positif
1480	Kerjasama dengan Astra, proses claim Reimburse...	1	negatif

1481 rows × 3 columns

Figure 6. Label of Score

The next stage was feature extraction using IndoBERT for text representation. In this stage, the pre-trained model indobenchmark/indobert-base-p1 was downloaded from HuggingFace and used to generate vector embeddings from each review. Figure 7 illustrates the process of loading various configuration and parameter files such as tokenizer_config.json, vocab.txt, and pytorch_model.bin (approximately 500 MB in size), which contains the model weights.

The text representation was obtained using the mean-pooling technique, where the hidden states of each token in a sentence are averaged to produce a single vector with a fixed dimension (768 features). Formally, this can be expressed as: The next stage was **feature extraction using IndoBERT** for text representation. In this stage, the pre-trained model **indobenchmark/indobert-base-p1** was downloaded from HuggingFace and used to generate vector embeddings from each review. Figure 7 illustrates the process of loading various configuration and parameter files such as *tokenizer_config.json*, *vocab.txt*, and *pytorch_model.bin* (approximately 500 MB in size), which contains the model weights.

The text representation was obtained using the **mean-pooling technique**, where the hidden states of each token in a sentence are averaged to produce a single vector with a fixed dimension (768 features). Formally, this can be expressed as:

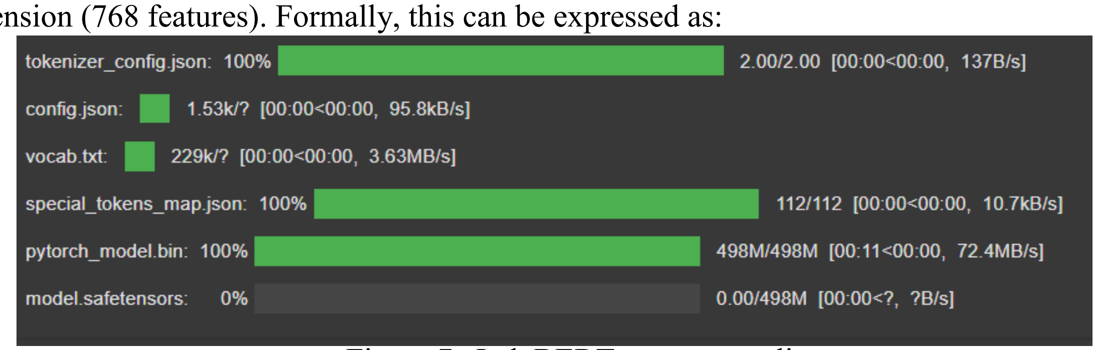

Figure 7. IndoBERT + mean-pooling

After the user review data was cleaned and labeled, the next stage was data splitting for model training and testing. This process was performed using the train_test_split function from scikit-learn, with a proportion of 80% for training data and 20% for testing data. The parameter random_state was set to a fixed value (SEED = 42) to ensure reproducibility, meaning that the experiment could be repeated with the same data partitioning.

In addition, the option stratify = y_rb was applied so that the distribution of sentiment labels in both the training and testing sets remained proportional to their original distribution. Formally, this can be expressed as:

```
# Split data
Xtr_txt, Xte_txt, ytr, yte = train_test_split(
    df_rb["clean_text"].tolist(), y_rb,
    test_size=0.2, random_state=SEED, stratify=y_rb
)
```

Figure 8. Split Data

After the Random Forest model was trained using IndoBERT representations, its performance was evaluated on the test data that had not been seen before. The evaluation results were presented in the form of a classification report, as shown in Figure 9. Overall, the model achieved an accuracy of 93%, with the highest F1-score in the Positive class (0.98). This finding aligns with the data distribution, where the majority of Halodoc user reviews tended to be positive.

For the Negative class, the F1-score was still fairly good at 0.67, with precision of 0.62 and recall of 0.73, indicating that the model was relatively capable of detecting negative reviews despite the limited number of samples. However, in the Neutral class, the model's performance was much lower, with an F1-score of only 0.14, which was influenced by the very small number of neutral reviews, making it difficult for the model to learn language patterns in this class.

The weighted average results demonstrated consistently high values for precision, recall, and F1-score (0.92–0.93), suggesting that the model was quite reliable overall. Nevertheless, the macro average was only around 0.60, indicating uneven performance across classes, particularly due to the weakness in the Neutral class.

Thus, although the model successfully delivered strong performance for the dominant class, further efforts such as data balancing or augmentation techniques are needed to improve the model's ability to classify all sentiment categories more evenly:

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

     negatif       0.62      0.73      0.67        22
      netral       0.33      0.09      0.14        11
     positif       0.97      0.98      0.98       264

    accuracy                           0.93       297
   macro avg       0.64      0.60      0.60       297
weighted avg       0.92      0.93      0.92       297
```

Figure 9. Classification Report

Figure 10 presents the model evaluation results in the form of a confusion matrix after the re-bucketing of sentiment labels. The confusion matrix provides a visual overview of the number of correct and incorrect predictions for each class. Based on the testing results, the model was able to predict the Positive class very well, with 260 positive reviews correctly classified. For the Negative class, 16 instances were correctly classified, although some misclassifications occurred, with 2 reviews predicted as Neutral and 4 reviews predicted as Positive. Meanwhile, the performance on the Neutral class remained low, with only 1 review correctly identified, while the rest were misclassified into the Negative class (6 reviews) or the Positive class (4 reviews).

In general, this pattern indicates that the model tends to have a bias toward the Positive class, which can be attributed to the fact that the majority of user reviews indeed come from users giving high scores. This leads to relatively weak performance in the Neutral class, due to the limited number of training samples in that category. Nevertheless, the confusion matrix also shows that the model is still fairly reliable in distinguishing Negative reviews from Positive ones, even though some misclassifications remain.

These results highlight the importance of applying data balancing strategies, such as SMOTE (Synthetic Minority Over-sampling Technique) or the inclusion of additional Neutral data, to improve the model's ability to classify all sentiment classes more evenly

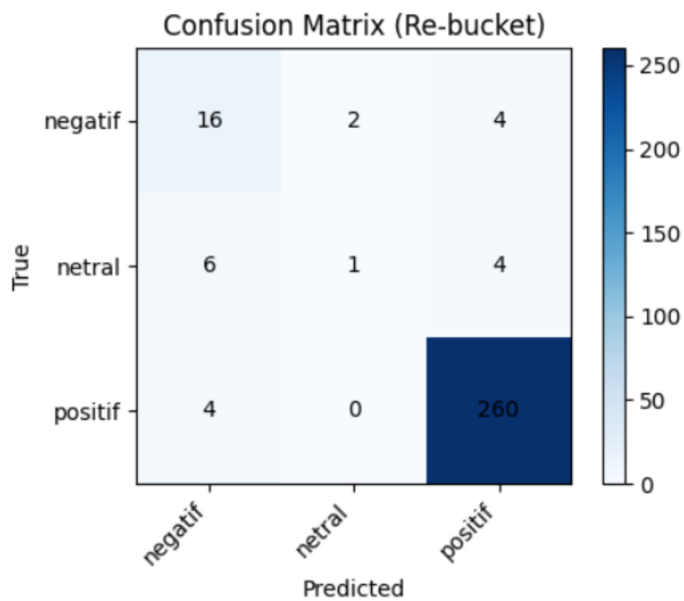

Figure 10. Confusion Matrix (Re-bucket)

Figure 11 shows a comparison of sentiment class distribution between the initial labeling results and after the re-bucketing process. In the Original Class Distribution chart, the Positive class was highly dominant with 1,315 reviews, while the Negative class contained only 138 reviews, and the Neutral class was very limited with only 28 reviews. This imbalance in distribution has the potential to cause the model to become biased toward the majority class, thereby reducing classification performance for the minority classes (Negative and Neutral).

To mitigate this imbalance, a re-bucketing strategy was applied by regrouping the rating scores: a score of 1 as Negative, scores of 2–3 as Neutral, and scores of 4–5 as Positive. The results can be seen in the Re-bucketed Class Distribution chart, where the number of Negative reviews became 111, Neutral reviews increased to 55, while Positive reviews remained 1,315. Although class imbalance still exists due to the dominance of the Positive class, the increase in Neutral reviews provides a more balanced dataset, which is expected to improve the model's ability to recognize Neutral sentiment variations.

This stage was essential as part of data adjustment before proceeding to the machine learning process using IndoBERT representation and the Random Forest algorithm.

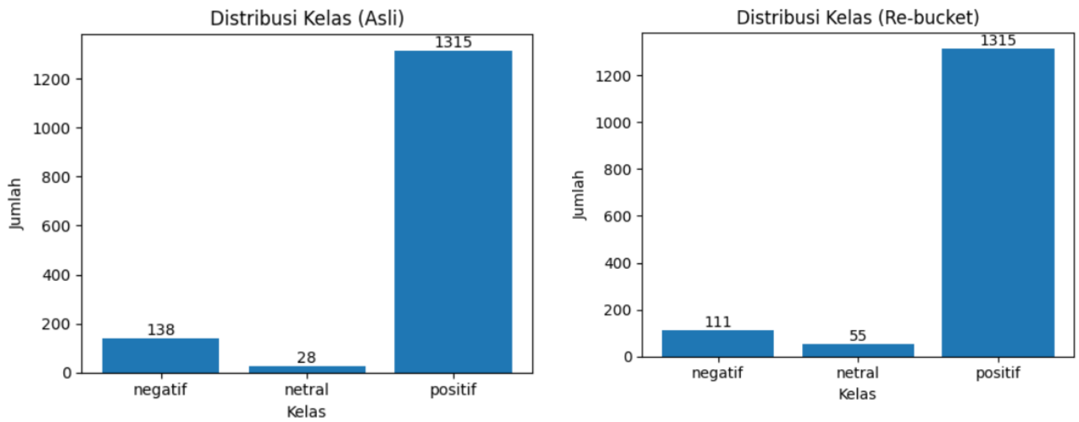

Figure 11. Distribution Graphic Distribution Class (Re-bucket)

As an additional output of this research, a **Flask-based web application prototype** was developed to enable interactive model testing. The application interface, as shown in Figure X, was designed to be simple yet functional, with two main features.
The first feature is the **Quick Test (Single Text)** menu, which allows users to input a single review sentence, after which the system immediately displays the predicted sentiment along with its confidence score. For example, the review *"Pelayanan apoteknya cepat dan ramah, obat lengkap"* (*"The pharmacy service is fast and friendly, with complete medicine"*) was successfully predicted as **Positive** with a confidence of **0.666**.

The second feature is **Batch via CSV**, where users can upload a CSV file containing multiple reviews with one text column (e.g., *clean_text*, *review*, or *content*). Once uploaded, the system processes the entire dataset simultaneously and generates an output file containing additional columns for sentiment predictions and confidence scores. Furthermore, the application also provides an **API endpoint (JSON)** that can be accessed by external systems for integration, such as with academic applications or monitoring dashboards.

The development of this prototype demonstrates that the **IndoBERT + Random Forest model** built in this study goes beyond analysis and can be implemented as a practical and user-friendly application. Thus, the outcomes of this research provide not only a **theoretical contribution** in terms of model performance evaluation but also a **practical contribution** in the form of a tool for real-time sentiment analysis of user reviews on the Halodoc pharmacy service

This research successfully developed and integrated two machine learning algorithms, Random Forest (RF) and XGBoost (XGB), into a web-based credit card fraud detection application using the Python Flask framework. The study demonstrated that both algorithms achieved high accuracy in detecting fraudulent transactions, with Random Forest showing stronger recall in identifying fraud cases, while XGBoost exhibited higher precision by minimizing false positives.

To enhance model performance in handling imbalanced data, SMOTE was applied during preprocessing, resulting in more balanced class distribution and improved detection of minority fraud cases. The models were further optimized through hyperparameter tuning to maximize accuracy and reliability.

The system was implemented as a functional Flask web application with key features including CSV upload for batch prediction, manual single-transaction testing, adjustable threshold settings, and downloadable prediction results. An additional ensemble approach (RF+XGB) was also provided, leveraging the complementary strengths of both algorithms to achieve more reliable fraud detection.

Overall, this research contributes to the development of a practical and effective fraud detection system that can be applied in real-time credit card transaction environments. Beyond academic significance, the proposed system has strong potential for adoption by financial institutions to mitigate risks of fraudulent activities and to strengthen digital payment security

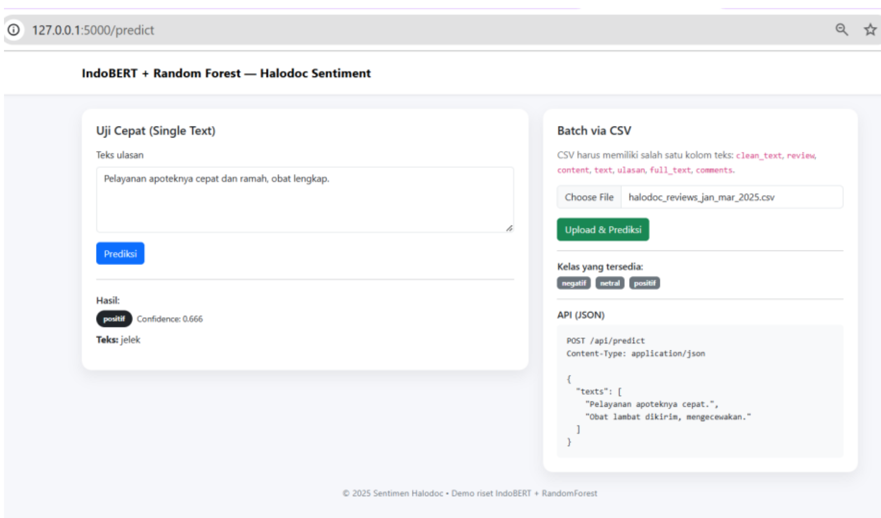

Figure 12. Prototype aplication web Flask Base

4. Conclusion

This research successfully demonstrated the integration of IndoBERT representations with the Random Forest algorithm for sentiment analysis of user reviews of the Halodoc pharmacy service on Google Play. The study was conducted through several structured stages, starting from data collection via web scraping, preprocessing and normalization of reviews, sentiment labeling, text representation with IndoBERT embeddings, model training and evaluation, and finally system implementation in a Flask-based web application.

The evaluation results showed that the proposed model achieved an overall accuracy of 93%, with the highest performance in the Positive class (F1-score of 0.98). Although the performance for the Negative class was reasonably good (F1-score of 0.67), the model still struggled to classify Neutral reviews (F1-score of 0.14) due to the limited number of neutral samples. The confusion matrix and class distribution analysis indicated a bias toward the Positive class, reflecting the real-world imbalance in user review data. The application of a re-bucketing strategy partially mitigated this issue, but further improvements such as data balancing techniques (e.g., SMOTE) and data augmentation are recommended to enhance the model's ability to classify all sentiment categories more evenly.

Beyond theoretical findings, this research also delivered a practical contribution by developing a Flask-based prototype application that allows both single-text and batch (CSV) testing, as well as API integration for external systems. This prototype highlights the applicability of the IndoBERT + Random Forest model as a practical sentiment analysis tool that can support decision-making and service improvement in digital pharmacy platforms like Halodoc.

In conclusion, the study contributes to both the academic and practical domains by demonstrating the effectiveness of combining IndoBERT embeddings with Random Forest for sentiment classification in Bahasa Indonesia, while also providing a functional tool that can be deployed in real-world scenarios.

References

[1]. Ardiansyah, Adika Sri Widagdo, Krisna Nuresa Qodri, Saputro FEN, Nisrina Akbar Rizky P. Analisis sentimen terhadap pelayanan Kesehatan berdasarkan ulasan Google Maps menggunakan BERT. J Fasilkom. 2023;13(02):326–33.

[2]. Geni L, Yulianti E, Sensuse DI. Sentiment Analysis of Tweets Before the 2024 Elections in Indonesia Using IndoBERT Language Models. J Ilm Tek Elektro Komput dan Inform [Internet]. 2023;9(3):746–57. Available from: http://journal.uad.ac.id/index.php/JITEKI
R. Nancy Deborh MSARGGGISJJJ. IRJET- Sentiment Analysis and Machine Learning Algorithm Implementation on Health Care System. Irjet. 2021;8(5):3466–71.

[3]. Fikry M, Oktavia L, Dwi Arista T, Islam Negeri Sultan Syarif Kasim Riau U, Soebrantas NoKm JH, Baru S. Klasifikasi Sentimen Masyarakat di Twitter terhadap Kenaikan Harga BBM dengan Metode K-NN Oleh : Tiara Dwi Arista Klasifikasi Sentimen Masyarakat di Twitter terhadap Kenaikan Harga BBM dengan Metode K-NN. JUKI J Komput dan Inform. 2023;5.

[4]. Desai A, Kalaskar S, Kumbhar O, Dhumal R. Cyber Bullying Detection on Social Media using Machine Learning. ITM Web Conf. 2021;40:03038.

[5]. Singgalen YA. Performance Analysis of IndoBERT for Sentiment Classification in Indonesian Hotel Review Data. 2025;6(2):976–86.

[6]. Manajemen J, Informasi S, Hazizah N, Feranika A, Komputer I, Ilmu F, et al. Implementasi Algoritma Random Forest Dalam Klasifikasi Risiko Gagal Bayar Kartu Kredit Pada Nasabah Bank Jurnal Manajemen Teknologi dan Sistem Informasi (JMS). 2025;5:1050–9.

[7]. Werdiningsih I, Purwanti E, Wira Aditya GR, Hidayat AR, Athallah RSR, Sahar VA, et al. Identifikasi Penipuan Kartu Kredit Pada Transaksi Ilegal Menggunakan Algoritma Random Forest dan Decision Tree. J Sisfokom (Sistem Inf dan Komputer). 2023;12(3):477–84.

[8]. Werdiningsih I, Purwanti E, Wira Aditya GR, Hidayat AR, Athallah RSR, Sahar VA, et al. Identifying Credit Card Fraud in Illegal Transactions Using Random Forest and Decision Tree Algorithms. J Sisfokom (Sistem Inf dan Komputer). 2023;12(3):477–84.

[9]. Pribadi MR, Manongga D, Purnomo HD, Setyawan I, Hendry. Sentiment Analysis of the PeduliLindungi on Google Play using the Random Forest Algorithm with SMOTE. 2022 Int Semin Intell Technol Its Appl Adv Innov Electr Syst Humanit ISITIA 2022 - Proceeding. 2022;(November):115–9.

[10]. N. Pohan, P. A. Widya Purnama, and H. Kurnia AR, “Pengenalan Aplikasi Teknologi Komputer untuk Edukasi Islami bagi Anak-anak TPQ Nurul Huda Gunung Pangilun, Kota Padang”, MB, vol. 1, no. 3, pp. 99–109, Sep. 2025, Accessed: Oct. 07, 2025. [Online]. Available: https://yasiinpublisher.org/index.php/mandalabakti/article/view/56

[11]. W. Akbar, “Sinergi Mahasiswa Prodi Ilmu Hukum Universitas Islam Kuantan Singingi dan Aparatur Penegak Hukum dalam Mendukung Efektivitas Pelayanan Publik pada Bagian Pemulihan Aset dan Barang Bukti (PAPBB)Kejaksaan Negeri Kuantan Singingi””, MB, vol. 1, no. 3, pp. 110–118, Sep. 2025, Accessed: Oct. 07, 2025. [Online]. Available: https://yasiinpublisher.org/index.php/mandalabakti/article/view/69

[12]. Febri Haswan, Nofri Wandi Al-Hafiz, A. Mualif, Afrinald Rizhan, Yudio Prastio, and Paris Paizal, “Pelatihan Optimalisasi Layanan Berbasis Teknologi: Aplikasi Antrian PTSP dan Sidang di Pengadilan Agama Teluk Kuantan”, MB, vol. 1, no. 3, pp. 119–125, Sep. 2025, Accessed: Oct. 07, 2025. [Online]. Available: https://yasiinpublisher.org/index.php/mandalabakti/article/view/57.

[13]. R. Gomez, F. Chen, and Y. Wang, “Human–AI collaboration for real-time skin cancer screening on smartphones,” *npj Digital Medicine*, vol. 7, no. 84, pp. 1–10, 2024.

[14]. J. K. Lee and S. H. Park, “Experimental AI research frameworks for medical image analysis,” *IEEE Access*, vol. 11, pp. 128221–128234, 2023.

[15]. M. Elgendi, C. Lovell, and D. Abbott, “Machine learning experimental design for healthcare applications,” *Sensors*, vol. 21, no. 22, p. 7551, 2021.

[16]. R. L. Mehta, P. Varghese, and K. Singh, “Designing user-centered interfaces for AI-powered healthcare applications,” IEEE Access, vol. 12, pp. 105321–105334, 2024. doi: 10.1109/ACCESS.2024.105321.

[17]. J. Kim, A. Kumar, and F. Torres, “Evaluating usability in mobile AI healthcare systems: A user-experience perspective,” International Journal of Human–Computer Studies, vol. 179, p. 103087, 2023. doi: 10.1016/j.ijhcs.2023.103087.

[18]. M. Chen, Z. Wang, and J. Xu, “Edge–cloud collaboration architecture for intelligent healthcare systems,” *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2134–2148, 2024.

[19]. H. Zhang, F. Li, and T. Nguyen, “A modular AI pipeline for mobile-based medical imaging applications,” *IEEE Access*, vol. 12, pp. 114521–114533, 2024.

[20]. S. Das, R. Banerjee, and K. Huang, “Deep learning infrastructure for real-time medical diagnostics in cloud–mobile ecosystems,” *ACM Transactions on Computing for Healthcare*, vol. 5, no. 1, pp. 1–15, 2023.

[21]. L. Martins, A. Costa, and R. Pereira, “Applying UML modeling in mobile health application development: A design-oriented approach,” *IEEE Access*, vol. 12, pp. 120453–120465, 2024.

[22]. T. Nguyen and S. Patel, “Model-driven engineering and UML-based design for AI-enabled mobile systems,” *Journal of Software Engineering and Applications*, vol. 17, no. 2, pp. 75–89, 2023.

[23]. A. Kumar, S. R. Gupta, and L. Zhang, “UML activity diagram modeling for AI-driven mobile healthcare applications,” *IEEE Access*, vol. 12, pp. 125231–125245, 2024.

[24]. R. Lopez and M. Fernandes, “Workflow modeling in mobile intelligent systems using UML activity diagrams,” *Journal of Systems and Software Engineering*, vol. 196, pp. 112015, 2023.

[25]. J. Fernandez, L. Silva, and P. R. Costa, “UML class modeling for modular AI-driven healthcare applications,” *IEEE Access*, vol. 12, pp. 130512–130528, 2024.