Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

Sentiment Analysis of ChatGPT App Reviews on the Play Store Using KNN and Decision Tree Methods

Hammam Aulia Nur Rahman*

School of Informatics, Informatics, Telkom University, Bandung, Indonesia
Email: [1,*] hammamaulia@students.telkomuniversity.ac.id,
Email Penulis Korespondensi: hammamaulia76@gmail.com
Submitted: **22/07/2025**; Accepted: **01/09/2025**; Published**: 04/09/2025**

Abstract−This study aims to analyze the sentiment of user reviews of the ChatGPT application on the Google Play Store, a platform that directly reflects public opinion toward this increasingly popular artificial intelligence application. A total of 10,000 reviews were collected through web scraping and underwent a series of rigorous preprocessing stages. These stages included data cleaning to remove noise, case folding to standardize text, tokenizing to break sentences into words, normalization to standardize informal words, and stopword removal to eliminate common but uninformative words—ensuring optimal data quality. Feature weighting was then performed using the Term Frequency-Inverse Document Frequency (TF-IDF) method with three n-gram scenarios (Unigram, Unigram+Bigram, Unigram+Trigram), followed by feature selection using Chi-Square to identify the most relevant features. The processed and weighted data were then classified using two machine learning algorithms: K-Nearest Neighbors (KNN) and Decision Tree. The evaluation results show that the Decision Tree model with Unigram+Bigram features achieved the highest accuracy of 0.8089 (80.89%) and an F1-Score of 0.8894 (88.94%), making it the best-performing model in this study. These findings provide valuable insights for application developers to better understand user perceptions, identify areas for improvement, and enhance the quality of ChatGPT services in the future, especially when addressing the challenge of imbalanced review data.

Keywords: Sentiment Analysis; ChatGPT; Decision Tree; KNN; TF-IDF; SMOTE

1. INTRODUCTION

In this era of rapid technological development, the application of artificial intelligence (AI) is becoming increasingly widespread. This technological development has led to the emergence of various applications based on natural language processing (NLP) [1]. One such application is ChatGPT. ChatGPT itself is a large language model developed by OpenAI in November 2022 [2] that has been utilized across numerous industries, businesses, and public services. ChatGPT responds to interactions between humans and machines using a deep learning-based artificial neural network model with 175 billion parameters, focusing on natural language processing [3]. The popularity of ChatGPT is also evident from the numerous user reviews available on app distribution platforms like the Google Play Store. According to download data from the Google Play Store, the ChatGPT app has been downloaded over 500 million times by June 2025, with a rating of 4.7.

User reviews on the Play Store are an important source of data that represent user satisfaction [4], complaints, and experiences when interacting with the app. The success of apps like ChatGPT can be directly seen in the reviews given by its users on the Play Store. Every review, whether positive or negative [5], contains information about the app's strengths, weaknesses, features that users like, and areas that need improvement. Understanding the sentiment contained in these reviews is key for developers to make data-driven decisions, prioritize feature development, and enhance the overall user experience.

Sentiment analysis, as an important branch of Natural Language Processing (NLP), is a field that focuses on identifying, extracting, and classifying opinions, emotions, or attitudes from text [6]. By applying sentiment analysis techniques to ChatGPT app reviews, we can determine whether users have a positive or negative view of the ChatGPT app. For example, reviews praising response speed or information accuracy will be categorized as positive, while complaints about bugs or confusing interfaces will be categorized as negative. Such insights are crucial in development to identify critical issues, measure customer satisfaction, and design more effective product improvement strategies.

Research in the field of sentiment analysis has been extensively conducted using various machine learning algorithms, including K-Nearest Neighbor (KNN) and Decision Tree. Farhan (2023) conducted sentiment analysis on ShopeeFood services using data from Twitter and compared the performance of three algorithms, namely KNN, SVM, and Decision Tree. The results of the study showed that SVM achieved the highest accuracy of 84.3%, followed by KNN at 81.5%, and Decision Tree at 78.2% [7]. This indicates that KNN is quite competitive as a text classification model.

Fitriani et al. (2022) studied public sentiment toward the implementation of the P3K teacher program using Naïve Bayes and Decision Tree. In that study, the decision tree achieved an accuracy of 79.31%, outperforming Naïve Bayes, which only reached 75.86% [8]. Meanwhile, Kusuma and Cahyono (2023) applied KNN to classify public sentiment toward the use of e-commerce and achieved an accuracy of 83.0% with k=3 [9]. These results reinforce the effectiveness of KNN in classification.

In the context of ChatGPT, Ramaputra and Purnomo (2024) studied public opinion on Twitter regarding the use of ChatGPT in education. The results showed that positive sentiment dominated at 61%, while negative and neutral

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

sentiment accounted for 25% and 14%, respectively [10]. Although this study did not measure the model's accuracy, the findings suggest that public opinion toward ChatGPT tends to be positive.

Cahyaningtyas et al. (2021) used a decision tree in analyzing Shopee app ratings. By applying data balancing techniques using SMOTE, the model achieved a high accuracy of 87.04% [11]. These findings indicate that decision tree performance can improve significantly when applied to balanced data. Meanwhile, Adhi Putra (2021) used the KNN algorithm to analyze reviews of the Bibit and Bareksa investment apps and achieved an accuracy of 80.4% with an F1-score of 0.79 [5].

A highly relevant study was conducted by Sagala and Samuel (2024), who analyzed user reviews of the ChatGPT app on the Google Play Store using the Random Forest, SVM, and Naïve Bayes algorithms. The results of this study showed that Random Forest had the highest accuracy of 91.23%, followed by SVM at 89.77%, and Naïve Bayes at 85.44% [2]. However, in this study, the KNN and decision tree algorithms were not tested, opening up the opportunity to compare the two algorithms in the same context. Astuti and Nuris (2022) also conducted sentiment analysis using KNN on reviews of the Peduli Lindungi application and achieved an accuracy of 78.5% [12]. This shows that the KNN algorithm can be relied upon to classify reviews of Indonesian-language applications.

In this study, the author used the K-Nearest Neighbors (KNN) and Decision Tree algorithms. These methods were chosen because KNN is a simple yet effective non-parametric algorithm that classifies new data based on its proximity to existing training data [9]. KNN itself has advantages in terms of ease of implementation and its ability to handle complex data distributions without specific model assumptions [13]. On the other hand, a decision tree is a tree-based algorithm that builds a series of decision rules based on data features [8]. Decision trees have several advantages, such as their ability to handle non-linear data, ease of interpretation, and their ability to identify the most important features in classification [14].

2. RESEARCH METHODOLOGY

2.1 Research Stages

In this study, a system was developed consisting of several layers to perform sentiment analysis on user reviews of the ChatGPT application. The system flow can be seen in Figure 1.

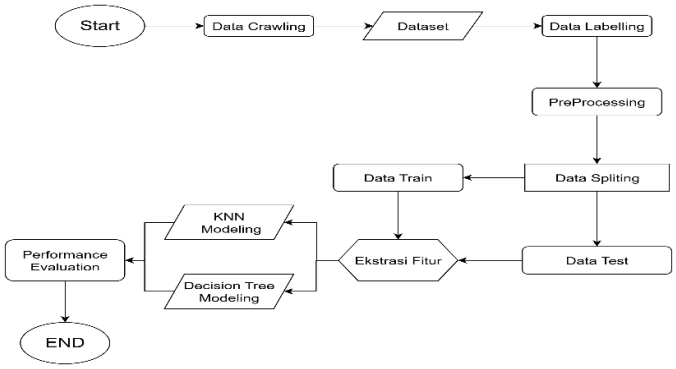

Figure 1. Flow System Design

The process flow in the image above illustrates the stages of sentiment analysis research using machine learning methods. The process begins with the data crawling stage, which involves collecting review data from sources such as the Google Play Store. The data obtained is then collected in the form of a dataset, followed by data labeling to assign sentiment labels such as positive, negative, or neutral. After that, preprocessing is performed to clean the text data of irrelevant elements, such as punctuation marks, capital letters, and so on. The next stage is data splitting, which involves dividing the data into two parts: training data for model training and test data for model testing. Before modeling, the processed text data must undergo feature extraction to convert it into a numerical format using techniques like TF-IDF. Then, modeling is performed using two classification algorithms: K-Nearest Neighbor (KNN) and Decision Tree. The results from both models are evaluated through a performance evaluation stage to measure model performance based on evaluation metrics like accuracy and F1-score.

2.2 Dataset

This study uses a dataset collected from ChatGPT app reviews on the Google Play Store, obtained through a crawling method [15] using the google-scrapper library. In this study, 10,000 app reviews were successfully collected. The data collected includes important information such as Review ID, User Name, Rating (star rating), Content (review content), and Timestamp (review time). After being successfully collected, all this data was then exported and saved in CSV file format with the name Hasil_scraper_ulasan_gpt.csv to facilitate further processing. Table 1 shows a summary of the scraping results.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

Table 1. Hasil Data *Crawling*

Review	Rating
Kemajuan teknologi ini memudahkan dan membantu, kalau dipakai sebaik mungkin buat kebaikan, karir, upgrade diri, dsb.	5
Jujur ini aplikasi paling nggak berguna yang pernah saya coba. Harapan saya tinggi, ternyata jawabannya ngawur, nggak akurat, kadang asal ngarang! Tanya hal yang simpel aja dijawab muter-muter.	1

2.3 Data Labeling

After the review data has been successfully collected, the next step is sentiment labeling. In this process, each review is labeled with a sentiment (positive (1) and negative (0)) based on the star rating system provided by users. Reviews with ratings of 1, 2, and 3 stars are automatically categorized as negative sentiment, indicating an unsatisfactory experience or opinion. Meanwhile, reviews with a rating of 4 or 5 stars are labeled as positive sentiment, indicating satisfaction or a good experience. From the sentiment-labeled dataset, there are 7,931 data points with positive labels and 2,069 data points with negative labels. This indicates an imbalance in the data. To address this, the SMOTE technique was used during the modeling stage to balance the training data by adding synthetic samples to the minority class. Further details are provided in the results and discussion section. The results of the data labeling can be seen in Table 2.

Table 2. Result Data *Labeling*

Review	Label
Kemajuan teknologi ini memudahkan dan membantu, kalau dipakai sebaik mungkin buat kebaikan, karir, upgrade diri, dsb.	1
Jujur ini aplikasi paling nggak berguna yang pernah saya coba. Harapan saya tinggi, ternyata jawabannya ngawur, nggak akurat, kadang asal ngarang! Tanya hal yang simpel aja dijawab muter-muter.	0

2.4 Pre-Processing

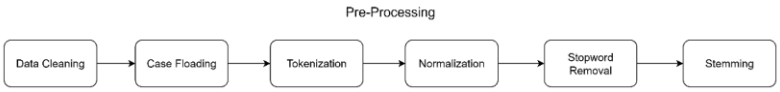

Figure 2. *Pre-Processing*

After data labeling, the next step is data preprocessing. Preprocessing aims to remove unnecessary data during the classification process to obtain better results [16]. In this study, there are several stages of preprocessing.

a. Data Cleaning
Data cleaning removes punctuation marks, hashtags, URLs, symbols, numbers, and empty attributes. It also corrects data inconsistencies and removes duplicate or redundant elements. This makes the data more structured and ready for analysis. Examples of punctuation marks, symbols, and numbers are shown in Table 3.

Table 3. Examples of Punctuation Marks, Symbols, and Numbers

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

b. Case Floading
Case folding is performed by converting all letters in the text to lowercase to standardize the data. This process facilitates analysis [17].

c. Tokenization
Tokenization breaks text into small units called words or tokens [18], usually using spaces as separators. This step aims to facilitate the process of analyzing or classifying text data.

d. Normalization
Text normalization involves identifying redundant words and replacing them with words from the KBBI (Big Indonesian Dictionary). To ensure accuracy, this process is carried out using a manually compiled dictionary and the Sastrawi library.

e. Stopward Removal
For stopword removal, words that are considered unimportant or common are discarded. Although these words usually have grammatical functions, they do not offer useful information for text analysis. This process improves the effectiveness of data analysis.

f. Stemming
The stemming process simplifies text for further analysis by converting inflected words into their base forms by removing their inflections.

Before conducting sentiment analysis, the collected data must undergo preprocessing to clean it up so that it is ready for further analysis. The results of the preprocessing can be seen in Table 4.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

Table 4. Pre-Processing Result

Steps	Review
App Review	Kemajuan teknologi ini memudahkan dan membantu, kalau dipakai sebaik mungkin buat kebaikan, karir, upgrade diri, dsb.
Data Cleaning	Kemajuan teknologi ini memudahkan dan membantu kalau dipakai sebaik mungkin buat kebaikan karir upgrade diri dsb
Case Folding	kemajuan teknologi ini memudahkan dan membantu kalau dipakai sebaik mungkin buat kebaikan karir upgrade diri dsb
Tokenization	['kemajuan', 'teknologi', 'ini', 'memudahkan', 'dan', 'membantu', 'kalau', 'dipakai', 'sebaik', 'mungkin, 'buat', 'kebaikan', 'karir', 'upgrade', 'diri', 'dsb']
Normalization	['kemajuan', 'teknologi', 'ini', 'memudahkan', 'dan', 'membantu', 'kalau', 'dipakai', 'sebaik', 'mungkin', 'buat', 'kebaikan', 'karir', 'upgrade', 'diri', 'dsb']
Stopword Removal	['kemajuan', 'teknologi', 'memudahkan', 'membantu', 'dipakai', 'sebaik', 'kebaikan', 'karir', 'upgrade', 'dsb']
Stemming	maju teknologi mudah bantu pakai baik karir upgrade dsb

2.5 Data Spliting

Data division is the process of dividing a collection of review data into two parts. In this study, the author performed data splitting with an 80:20 ratio. Where 80% is training data [13] for training the classification model (KNN and decision tree), and 20% is test data for objectively evaluating the model's performance on previously unseen data, thereby preventing overfitting and providing accurate results. The results of the data splitting can be seen in Table 5.

Table 5. Jumlah Data *Train* Dan Data *Test*

Data Training	7992
Data Test	1999

2.6 Feature Weighting

The method used for feature weighting in this study is Term Frequency-Inverse Document Frequency (TF-IDF) [12]. TF-IDF is a statistical technique that reflects how important a word is in a document relative to the document corpus. The TF-IDF weight increases in proportion to the number of times a word appears in a document but is balanced by the frequency of that word across the entire corpus, which helps control for the fact that some words appear more frequently in general.

Feature weighting is performed with three different n-gram scenarios to evaluate the influence of word context on model performance:

a. Unigram: In this scenario, each single token (word) from the review is considered an independent feature. This is the most basic approach and captures the frequency of individual words.
b. Unigram + Bigram: This scenario combines single tokens (unigrams) with pairs of consecutive tokens (bigrams). Bigrams are important because they can capture phrases or word combinations that have different sentiment meanings from their individual words (e.g., “not good” vs. “good”).
c. Unigram + Trigram: This scenario expands on the previous approach by including single tokens, pairs of consecutive tokens (bigrams), and three consecutive tokens (trigrams). Trigrams can capture more complex contexts, although they are often less frequent and can significantly increase feature dimension.

2.7 K-Nearest Neighbor (KNN)

K-Nearest Neighbor is an instance-based learning classification algorithm, where the classification of new data is determined based on its similarity to the closest training data. KNN is known to be simple yet quite effective in text classification, including sentiment analysis [19]. This algorithm does not perform explicit training but rather stores all training data and determines the class based on the majority class of its closest neighbors. The distance calculation commonly used in KNN is the Euclidean distance:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (1)$$

where p and q are feature vectors of two different data points.

In text classification, such as sentiment analysis, text data is first converted into numerical vectors using representation techniques such as TF-IDF [20]. KNN is suitable for use because it has competitive accuracy and simple implementation, although it has weaknesses in terms of efficiency when the amount of data is large.

2.8 Decision Tree

A decision tree is a classification algorithm that works by dividing data into branches based on specific feature values. The main advantage of a decision tree is its ability to provide a visual interpretation of the classification process [21]. A decision tree is also a classification algorithm that divides data into a decision tree structure based on specific feature

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

values. This tree is formed recursively, with each internal node representing a test of an attribute and each branch showing the result of that test. The attribute selection process typically uses the Gini Index formula:

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2 \quad (2)$$

where p_i is the proportion of elements in class i. Decision trees are widely used in text classification because the model results are easy to understand and visualize [22]. In addition, this model is capable of handling large amounts of labeled data and does not require much pre-processing.

2.9 *Evaluasi Confuison Matrix*

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

Table 6. *Confuision Martix*

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

For a more in-depth analysis of model performance, the matrix helps in calculating evaluations such as accuracy, precision, recall, and F1-score.
Accuracy: Measuring the proportion of correctly classified data against the entire data set.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (4)$$

Precision: Measuring the accuracy of positive class predictions compared to all positive predictions.

$$Precission = \frac{TP}{(TP+FP)} \quad (5)$$

Recall: Measuring the model's ability to find all truly positive data.

$$Recall = \frac{TP}{(TP+FN)} \quad (6)$$

F1-Score: The harmonic mean of precision and recall provides a balanced view of both.

$$F1 - Score = \frac{Precission \times Recall}{(Precission+Recall)} \quad (7)$$

3. RESULTS AND DISCUSSION

3.1 WordCloud Visualization Results

In this study, visual analysis was conducted using Word Cloud based on positive and negative sentiments from user reviews. This visualization aims to identify the words that appear most frequently in each sentiment category, thereby providing an overview of the aspects that users appreciate and complain about. With this approach, certain patterns in user opinions can be identified more intuitively and informatively.

Figure 3. WordCloud Visualization Results

The image above shows a word cloud visualization depicting the distribution of words that frequently appear in user reviews of the ChatGPT application, separated by positive sentiment (left) and negative sentiment (right). In the positive sentiment word cloud, words such as “good,” “helpful,” “ChatGPT,” “application,” and “thank you” dominate, indicating that many users find the application helpful, perform well, and deserve appreciation. These positive reviews reflect user satisfaction with the application's ease of use, response speed, and benefits for learning or work.

Meanwhile, the word cloud for negative sentiment shows a dominance of words like “really,” “use,” “limit,” “image,” “photo,” and “update.” This indicates complaints from users regarding app usage limits, technical issues such as errors or bugs, and certain features not functioning as expected, particularly regarding image uploads or app updates. Words like “please” also indicate users' requests or hopes that these issues will be resolved promptly.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

3.2 Evaluation Results Using KNN

The performance of the K-Nearest Neighbors (KNN) model was evaluated to classify the sentiment of user reviews of the ChatGPT application. Before the evaluation, the KNN model was first built using data that had undergone preprocessing and TF-IDF feature weighting. The main parameters used were the number of nearest neighbors (k) and the calculation of distance using Euclidean distance. The model was trained using 80% of the dataset as training data and tested on the remaining 20%. The testing included three text feature representation scenarios: unigram, unigram+bigram, and unigram+trigram.

The performance of the K-Nearest Neighbors (KNN) model was evaluated to classify the sentiment of user reviews of the ChatGPT application. The testing included three text feature representation scenarios: unigram, unigram+bigram, and unigram+trigram. The analysis used classification metrics such as precision, recall, F1 score, and accuracy, as well as confusion matrix visualization to assess the model's ability to distinguish between positive and negative sentiments. The evaluation results provided an overview of the effectiveness of each feature representation on classification performance.

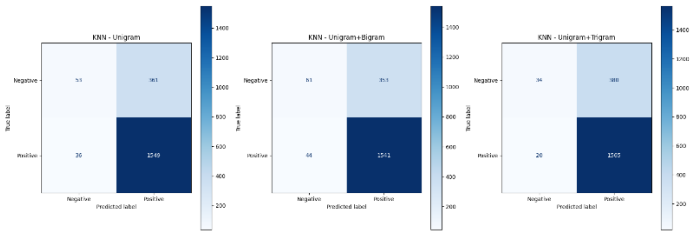

Figure 4. *Confusion Matrix* KNN

The confusion matrix results in Figure 4 show that in the unigram scenario, the model is quite good at recognizing positive sentiment (1,549 true positives), but still weak at recognizing negative sentiment (only true negatives). With the addition of bigrams, negative classification improves (61 true negatives), although positive predictions decline slightly. In the trigram scenario, positive predictions are very high (1,565 True Positives), but the ability to recognize negative reviews decreases drastically (34 True Negatives), indicating that the model is too focused on the positive class.

Table 7. *Clasification Report KNN*

Model	Sekenario	Precision	Recall	F1-Score	Accuracy
KNN	Unigram	76.09%	79.94%	74.39%	79.94%
	Unigram + Bigram	76.88%	80.29%	75.44%	80.29%
	Unigram + Trigram	77.91%	80.19%	73.44%	80.19%

Table 7 provides further details on the model's performance. In the unigram scenario, the model's accuracy was 79.94% with a precision of 76.09%, a recall of 79.94%, and an F1-score of 74.39%. The relatively low F1 score compared to precision and recall indicates an imbalance in performance between classes. When bigrams are added (unigram+bigram), all metrics improve. Accuracy increases to 80.29%, precision to 76.88%, recall remains at 80.29%, and the F1-score improves to 75.44%. This indicates that adding bigrams provides additional contextual information that is beneficial to the model. In the unigram+trigram combination, the model's precision is the highest at 77.91%, but the F1-score decreases slightly to 73.44%, indicating a trade-off between precision and recall. Overall, the KNN model with unigram+bigram representation provides the most balanced performance in terms of precision and recall, while trigrams tend to make the model overly confident in classifying positive sentiment but sacrifice accuracy for negative classes.

Over-sampling Technique (SMOTE), after which a re-evaluation of the performance of the K-Nearest Neighbors (KNN) model applied to the training data was conducted before the model training process was carried out. This technique increases the data in the minority (negative) class by synthesizing it based on the proximity of features from the existing data. This evaluation aims to see the extent of the model's improved ability to classify both sentiment classes evenly. The assessment is still carried out using a confusion matrix and classification report on three types of feature representations: unigram, unigram+bigram, and unigram+trigram.

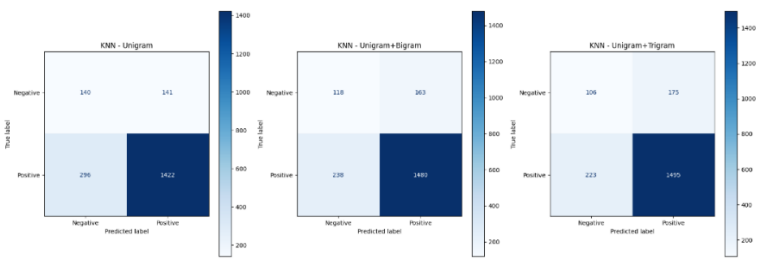

Figure 5. *Confusion Matrix* KNN After *SMOTE*

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

The confusion matrix shows that the KNN model is very good at recognizing positive reviews, especially in trigram representation (TP: 1,495), but weaker at detecting negative reviews (TN: only 106). For unigrams, the model recorded a TP of 1.422 and a TN of 140, while for bigrams, the TP increased to 1.480 and the TN decreased to 118. In general, as the complexity of the features used increases, the model tends to become more biased toward the positive class, with an increase in false positives and a decrease in the accuracy of negative sentiment classification.

Model	Skenario	Accuracy	Precision	Recall	F1-Score
KNN	Unigram	0.7814	0.8270	0.7814	0.7999
KNN	Unigram+Bigram	0.7994	0.8208	0.7994	0.8090
KNN	Unigram+Trigram	0.8009	0.8147	0.8009	0.8073

Figure 6. *Clasification report KNN* After *SMOTE*

The classification report above shows that the performance of the KNN model gradually improves with the addition of features, from an F1-score of 0.7999 (unigram) to 0.8090 (bigram) to 0.8073 (trigram), with the highest accuracy of 80.09% on trigrams. Although the evaluation values appear to be improving, the imbalance in detecting negative classes remains a concern. This indicates that improved accuracy does not always translate into the model's ability to recognize both classes equally.

3.3 Evaluation Results Using Decision Tree

In addition to the KNN model, this study also examines the performance of the Decision Tree (DT) model in the same scenario. The decision tree was chosen because of its interpretable nature and frequent use in classification tasks. The decision tree was implemented using a data separation algorithm based on the Gini Index, and the model was built recursively based on features generated from TF-IDF. The data used were preprocessed and divided into training and testing datasets. To address overfitting, the maximum tree depth parameter and the minimum number of samples per leaf were set to optimal values based on initial experiments. Similar to KNN, the DT model was tested under imbalanced data conditions and after data balancing using SMOTE. Evaluation was conducted to assess the model's ability to generalize sentiment patterns from the training data to the testing data.

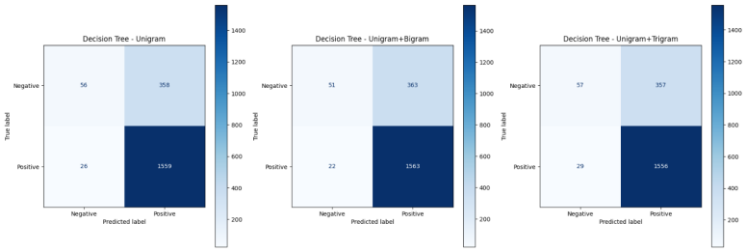

Figure 7. *Confusion Matrix* DT

Based on the confusion matrix results for the decision tree model in Figure 7, it shows that the model is very dominant in recognizing positive classes. In the unigram representation, the model successfully predicted 1,559 positive data correctly (true positive) and 56 negative data correctly (true negative). In the unigram+bigram scenario, the number of true positives slightly increased to 1,563, although true negatives decreased to 51. Meanwhile, in the unigram+trigram scenario, the number of true positives slightly decreased to 1,556, and true negatives increased to 57. Overall, the model tends to overfit the positive class, while its ability to detect the negative class remains low.

Table 8. *Clasificatiaon Report* DT

Model	Scenario	Precision	Recall	F1-Score	Accuracy
Decision Tree	Unigram	78.04%	80.59%	75.01%	80.59%
	Unigram + Bigram	78.37%	80.59%	74.71%	80.59%
	Unigram + Trigram	78.81%	80.84%	75.31%	80.84%

Based on classification evaluation metrics, the decision tree model showed relatively consistent performance across all three feature representation scenarios. Accuracy scores ranged from 74.71% to 75.31%, with F1-scores around 80%. Although the differences between scenarios are not significant, the scenario with unigram+trigram produced the highest accuracy (75.31%), while unigram+bigram recorded the highest F1-score (80.89%). This shows that the addition of bigrams or trigrams does not provide a significant improvement to the performance of the decision tree.

Next, data imbalance was addressed using the Synthetic Minority Over-sampling Technique (SMOTE), which was applied to the training data before the model training process was carried out. This technique increases the data in the minority class (negative) synthetically based on the feature proximity of the existing data. After this oversampling process, the decision tree model was retrained and tested with the same test data to evaluate the impact of SMOTE on model performance.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

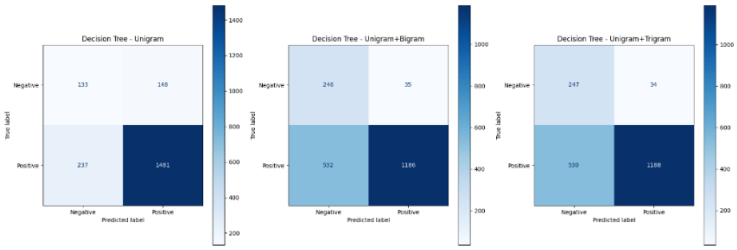

Figure 8. *Confusion Matrix* DT After SMOTE

After balancing with SMOTE, the decision tree's performance showed significant changes in negative sentiment classification. In the unigram representation, True Positive (1.481) and True Negative (133) showed an increase in the model's ability to recognize both classes, although it was still quite biased towards the positive class. However, in the unigram+bigram and unigram+trigram representations, the model begins to show better balance, with True Negative reaching 246 and 247, respectively, while True Positive remains high at over 1,180. This indicates that SMOTE helps improve the detection of the minority class.

Model	Skenario	Accuracy	Precision	Recall	F1-Score
Decision Tree	Unigram	0.8074	0.8319	0.8074	0.8180
Decision Tree	Unigram+Bigram	0.7164	0.8792	0.7164	0.7589
Decision Tree	Unigram+Trigram	0.7179	0.8802	0.7179	0.7602

Figure 9. *Clasification report* DT after SMOTE

From the evaluation table, the highest accuracy was achieved by unigrams (80.74%), but the unigram+trigram representation provided more balanced precision and recall (precision 0.8802, recall 0.7179). This indicates that balancing through SMOTE not only improves recall for the minority class but also maintains precision in the majority class, especially when using n-gram feature combinations.

3.4 Comparison of Results

To gain a deeper understanding of the performance of the classification model used in this study, an evaluation was conducted on two machine learning algorithms, namely K-Nearest Neighbors (KNN) and Decision Tree (DT). This evaluation was based on several key performance metrics, such as precision, recall, F1 score, and accuracy. Additionally, the evaluation also considered variations in text feature representation using N-gram schemes, namely unigram, unigram+bigram, and unigram+trigram. These metrics provide a comprehensive overview of the model's effectiveness in classifying review text data. The following graph visualization displays a comparison of the performance of the two models on each N-gram scheme, allowing for analysis of which model consistently outperforms the other on each metric. Figure 10 shows a comparison graph between KNN and Decision Tree in the N-Gram scheme.

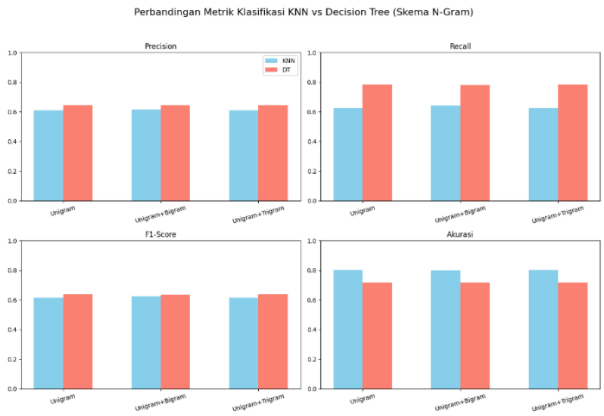

Figure 10. Comparison KNN and DT

The figure above shows a comparison of classification performance metrics between the K-Nearest Neighbors (KNN) and Decision Tree (DT) algorithms after handling data imbalance using SMOTE based on the N-Gram text feature extraction scheme, namely Unigram, Unigram+Bigram, and Unigram+Trigram. There are four main metrics compared: precision, recall, F1-score, and accuracy.

In terms of precision, the Decision Tree algorithm consistently achieves higher values than KNN across all N-Gram schemes, although the differences are relatively small. This indicates that DT tends to be more accurate in predicting positive classes with fewer false positives. Meanwhile, in the recall metric, DT outperforms KNN significantly, with DT recall values reaching 0.79 in all schemes, while KNN only ranges from 0.62 to 0.64. This

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

indicates that DT has a much better ability to detect all positive class instances, which is very important in the context of classification that focuses on sensitivity.

The F1-score, as a combination of precision and recall, shows that DT's performance remains slightly superior to KNN, although the difference is not very significant. KNN recorded an F1-score between 0.62 and 0.63, while DT remained stable at 0.64 in all three schemes. However, in terms of accuracy metrics, KNN actually performs better than DT, with a stable accuracy value of 0.80, while DT only reaches 0.72 across all N-Gram schemes. This indicates that, overall, KNN more frequently provides correct predictions; however, this may be due to the dominance of the majority class, so the accuracy metric can be misleading if not analyzed alongside other metrics.

In general, the performance of both models is relatively stable across the three text feature schemes used, so the addition of bigram and trigram does not have a significant effect on the evaluation results. From these results, it can be concluded that Decision Tree is superior in terms of sensitivity (recall) and fairly balanced in precision and f1-score, making it more suitable for classification tasks that require comprehensive detection. Conversely, KNN is suitable for use when the main focus is overall accuracy. The selection of the best model must still be tailored to the purpose and context of the algorithm's use in the research.

The application of SMOTE has different effects on the two models. In KNN, SMOTE improves the F1-score and recall, but the model still tends to be biased toward the positive class. Pen
The application of SMOTE had different effects on the two models. In KNN, SMOTE improves the F1-score and recall, but the model still tends to be biased toward the positive class. The improvement in performance for the negative class is not significant. Meanwhile, in Decision Tree, SMOTE has a stronger impact: recall for the negative class increases sharply, precision remains high, and the classification distribution becomes more balanced. Thus, SMOTE is more effective in decision trees in addressing data imbalance compared to KNN.

4. CONCLUSION

This study analyzes the sentiment of 10,000 ChatGPT app reviews from the Google Play Store. The analysis process involves rigorous data preprocessing and TF-IDF feature weighting with n-gram scenarios (unigram, unigram+bigram, and unigram+trigram), followed by classification using the K-Nearest Neighbors (KNN) and Decision Tree algorithms. Before applying the Synthetic Minority Over-sampling Technique (SMOTE), the Decision Tree model with Unigram+Bigram features achieved the highest accuracy of 0.8089 (80.89%). Meanwhile, the decision tree with unigram features produced the best F1 score of 0.8904 (89.04%). In general, the decision tree outperformed KNN. The use of bigrams proved effective in capturing complex sentiment contexts, but the addition of trigrams tended to reduce model performance due to potential noise or overfitting. After applying SMOTE, the KNN model with Unigram+Trigram features achieved the highest accuracy of 0.8009 (80.09%), and the Decision Tree with Unigram features produced the highest F1-Score of 0.8180 (81.80%). Although SMOTE successfully balanced the class distribution and improved the ability to identify minority classes, the highest overall F1-score (0.8904) was still achieved by the decision tree with unigram before SMOTE. This indicates that the impact of SMOTE on overall performance metrics needs to be carefully evaluated, as it does not always result in significant improvements across all metrics. For future research, it is recommended to explore more advanced feature weighting methods such as word embeddings (e.g., Word2Vec, GloVe) or contextual embeddings from pre-trained language models (e.g., BERT, FastText). Additionally, considering the use of deep learning models (such as recurrent neural networks or convolutional neural networks) could also be a promising research direction for more in-depth and accurate sentiment analysis in the future.

REFERENCES

[1] D. Trihapningsari, A. Widyasuri, M. A. Putri, dan A. Fatihin, “Sentiment Analysis of ChatGPT Exploration Based on Opinions on Platform X Using Naïve Bayes Algorithm,” *The 4th International Seminar of Science and Technology (ISST 2024)*, vol. 4, Art. no. 009, pp. 94–101, 2025. [Online]. Tersedia: http://jurnal.ut.ac.id/isst

[2] G. Jeffson Sagala dan Y. T. Samuel, “Sentiment Analysis on ChatGPT App Reviews on Google Play Store Using Random Forest Algorithm, Support Vector Machine and Naïve Bayes,” *Int. J. Eng. Bus. Soc. Sci.*, vol. 2, no. 04, hlm. 1194–1204, Mar 2024, doi: 10.58451/ijebss.v2i04.148.

[3] A. S. Pamungkas dan N. Cahyono, “Analisis Sentimen Review ChatGPT di Play Store menggunakan Support Vector Machine dan K-Nearest Neighbor,” *Edumatic J. Pendidik. Inform.*, vol. 8, no. 1, hlm. 1–10, Jun 2024, doi: 10.29408/edumatic.v8i1.24114.

[4] F. A. Larasati, D. E. Ratnawati, dan B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 9, hlm. 4305–4313, Sep. 2022. [Online]. Tersedia: http://j-ptiik.ub.ac.id

[5] A. D. Adhi Putra, “Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN,” *JATISI J. Tek. Inform. Dan Sist. Inf.*, vol. 8, no. 2, hlm. 636–646, Jun 2021, doi: 10.35957/jatisi.v8i2.962.

[6] A. Sabir, H. A. Ali, dan M. A. Aljabery, “ChatGPT Tweets Sentiment Analysis Using Machine Learning and Data Classification,” *Informatica*, vol. 48, no. 7, Mei 2024, doi: 10.31449/inf.v48i7.5535.

[7] M. Z. Farhan, “Analisis Sentimen Layanan Shopeefood Pada Twitter Dengan Metode K-Nearest Neighbor, Support Vector Machine, dan Decision Tree,” *J. Ilm. Inform.*, vol. 7, no. 2, hlm. 95–106, Jan 2023, doi: 10.35316/jimi.v7i2.95-106.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1242−1251
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8078

[8] Fitriani Fitriani, Ema Utami, dan Anggit Dwi Hartanto, "Analisis Sentimen Masyarakat Terhadap Pelaksanaan P3k Guru Dengan Algoritma Naive Bayes Dan Decision Tree," *Tek. Teknol. Inf. Dan Multimed.*, vol. 3, no. 1, hlm. 23–30, Jun 2022, doi: 10.46764/teknimedia.v3i1.53.

[9] I. H. Kusuma dan N. Cahyono, "Analisis Sentimen Masyarakat Terhadap Penggunaan E-Commerce Menggunakan Algoritma K-Nearest Neighbor," *J. Inform. J. Pengemb. IT*, vol. 8, no. 3, hlm. 302–307, Sep 2023, doi: 10.30591/jpit.v8i3.5734.

[10] M. G. Ramaputra dan H. Purnomo, "Analisis Sentimen Opini Masyarakat Terhadap Penggunaan ChatGPT di Bidang Pendidikan Berbasis Twitter," *J. Pepadun*, vol. 5, no. 3, hlm. 275–285, Des 2024, doi: 10.23960/pepadun.v5i3.242.

[11] C. Cahyaningtyas, Y. Nataliani, dan I. R. Widiasari, "Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE," *AITI*, vol. 18, no. 2, hlm. 173–184, Nov 2021, doi: 10.24246/aiti.v18i2.173-184.

[12] P. Astuti dan N. Nuris, "Penerapan Algoritma KNN Pada Analisis Sentimen Review Aplikasi Peduli Lindungi," *Comput. Sci. CO-Sci.*, vol. 2, no. 2, hlm. 137–142, Jul 2022, doi: 10.31294/coscience.v2i2.1258.

[13] M. Alidin dan R. Fadilah, "Optimasi KNN dengan PSO untuk Klasifikasi Kasus Hukum di Australia Menggunakan N-Gram," *SEIS: Seminar Nasional Sistem Informasi*, vol. 5, no. 1, pp. 26–34, Jan. 2025. [Online]. Tersedia: https://ejurnal.umri.ac.id/index.php/SEIS

[14] A. Fatkhudin, F. A. Artanto, N. A. Safli, dan D. Wibowo, "Decision Tree Berbasis SMOTE dalam Analisis Sentimen Penggunaan Artificial Intelligence untuk Skripsi," *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 8, no. 2, Apr. 2024, doi: 10.33395/remik.v8i2.13531.

[15] T. Y. Pahtoni dan H. Jati, "Analisis Sentimen Data Twitter Terkait Chatgpt Menggunakan Orange Data Mining," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 11, no. 2, hlm. 329–336, Apr 2024, doi: 10.25126/jtiik.20241127276.

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

[17] S. AL-Hagree, G. Al-Gaphari, F. H. Abdulrazzak, M. Al-Sanabani, dan A. Al-Shalabi, "Using Machine Learning for Arabic Sentiment Analysis in Higher Education: Investigating the Impact of Utilizing the ChatGPT and Bard Google," *J. Eng. Technol. Sci. - JOEATS*, vol. 3, no. 1, hlm. 9–23, Mar 2025, doi: 10.59421/joeats.v3i1.2473.

[18] R. Alawaji dan A. Aloraini, "Sentiment Analysis of Digital Banking Reviews Using Machine Learning and Large Language Models," *Electronics*, vol. 14, no. 11, hlm. 2125, Mei 2025, doi: 10.3390/electronics14112125.

[19] M. Nanda Fahriza dan N. Riza, "Analisis Sentimen Pada Ulasan Aplikasi Chat Generative Pre-Trained Transformer Gpt Menggunakan Metode Klasifikasi K-Nearest Neighbor(KNN): Sistematic Literature Review," *JATI J. Mhs. Tek. Inform.*, vol. 7, no. 2, hlm. 1351–1358, Sep 2023, doi: 10.36040/jati.v7i2.6767.

[20] Syahril Dwi Prasetyo, Shofa Shofiah Hilabi, dan Fitri Nurapriani, "Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN," *J. KomtekInfo*, vol. 10, no. 1, hlm. 1–7, Jan 2023, doi: 10.35134/komtekinfo.v10i1.330.

[21] R. Puspita dan A. Widodo, "Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS," *J. Inform. Univ. Pamulang*, vol. 5, no. 4, hlm. 646, Des 2021, doi: 10.32493/informatika.v5i4.7622.

[22] R. Fatmasari, V. M. Ayu, H. Anto, W. Gata, dan L. D. Yulianto, "Analisis Sentimen Dalam Pengkategorian Komentar Youtube Terhadap Layanan Akademik dan Non-Akademik Universitas Terbuka Untuk Prediksi Kepuasan," *Build. Inform. Technol. Sci. BITS*, vol. 4, no. 2, hlm. 395–404, Sep 2022, doi: 10.47065/bits.v4i2.1738.