Comparison of The Accuracy of K-Nearest Neighbor and Roberta Algorithm in Analysis of Sentiment on Miawaug Youtube Channel Comments

Fachrudin Okta Rahmawan[1*], Hanafi[2], Windha Mega Pradnya Dhuita[3]

[1]Informatics Study Program, Universitas Amikom Yogyakarta, Sleman, DI Yogyakarta, Indonesia
[2,3]Master in Informatics Engineering, Universitas Amikom Yogyakarta, Sleman, DI Yogyakarta, Indonesia
E-mail: [1*]fachrudinokta@students.amikom.ac.id, [2]hanafi@amikom.ac.id, [3]windha@amikom.ac.id

(Received: 30 Oct 2024, revised: 13 Nov 2024, accepted: 14 Nov 2024)

Abstract

This study aims to evaluate the accuracy of two algorithms, K-Nearest Neighbor (KNN) and Robustly Optimized BERT Approach (RoBERTa), in analyzing sentiment within comments on MiawAug's YouTube channel. Sentiment analysis was conducted on two sentiment categories: binary classification (positive and negative) and multi-class classification (positive, neutral, and negative). Using KNN, the binary classification yielded an accuracy of 86.12%, F1-score of 87.44%, recall of 96.64%, and precision of 79.89%. In contrast, the multi-class classification achieved 98.21% accuracy, F1-score, and recall with a precision of 98.23%. However, the RoBERTa model outperformed KNN, achieving 93.89% accuracy, 93.88% F1-score, 94.59% recall, and 93.22% precision in binary classification. For multi-class classification, RoBERTa further excelled, attaining 99.21% across accuracy, F1-score, recall, and precision. These findings demonstrate that RoBERTa surpasses KNN in sentiment analysis, especially in multi-class contexts, indicating its greater robustness for this application.

Keywords: Sentiment Analysis, K-Nearest Neighbor (KNN), RoBERTa, YouTube Comments.

I. INTRODUCTION

In the ever-evolving digital era, sentiment analysis has become an essential tool for understanding public opinions and perceptions regarding various content available on the internet. YouTube, as one of the largest video platforms in the world, provides a space for millions of users to express their opinions through comments [1]. This has also given rise to a new profession: the YouTuber. This job may not have been recognized by society before the advent of the internet. Content creators often upload their works with the aim of generating income from advertisements played on their videos [2], [3], [4]. Certainly, the YouTuber profession should not be underestimated, as the income they can earn can be quite significant. According to Social Blade, a YouTuber with 1 million subscribers can earn a monthly income ranging from IDR 38,548,320 to IDR 618,379,300.

With such earning potential, an increasing number of people aspire to become content creators in the future. One example is the gaming channel MiawAug, which has amassed over 20 million subscribers and has become one of the most liked channels, especially in Indonesia, despite its unique name. One of MiawAug's strengths, whose real name is Reggie Prabowo Wongkar, is that he never uses profanity while playing video games.

As MiawAug's popularity rises, there is an increasing need for a deeper understanding of the mood and public opinion regarding the content shared. Sentiment analysis is a method used to discover and categorize feelings and viewpoints in text. One of the methods frequently employed is through machine learning algorithms. In this context, the use of machine learning algorithms for sentiment analysis becomes highly relevant. Two popular methods used are the K-Nearest Neighbor (K-NN) algorithm and RoBERTa (Robustly Optimized BERT Approach) [5], [6]. K-NN is known for its ease of use and efficiency in clustering data based on proximity to previous data. On the other hand, RoBERTa, which is a variant of the BERT model, excels in understanding context and nuances of natural language through a more thorough and efficient training process [7].

This research aims to compare the accuracy of both algorithms in analyzing sentiment in comments on the MiawAug YouTube channel. This channel was chosen due to its extensive fan interactions and extraordinary popularity, with 22.8 million subscribers providing sufficient data for in-depth analysis. Moreover, the variety of comments on this channel, ranging from support to criticism, makes it a perfect subject for sentiment analysis research.

The research methodology includes collecting comment data from several videos on the MiawAug YouTube channel. The K-NN and RoBERTa algorithms will be used to process and analyze this data. This process involves data preprocessing, model training, and accuracy evaluation to determine which method is more effective in categorizing sentiment in comments.

One of the biggest challenges in sentiment analysis is the preprocessing stage. YouTube comment collections often contain slang and stop words. Words such as *"bagaimanapun juga", "atau", "dari", "tetapi",* "dan" are examples of stop words, which are words that do not carry sentiment. In Indonesian grammar, there are sixteen types of conjunctions. Additionally, stop words also include pronouns, adverbs of time, prepositions, and other words that do not provide significant information. Therefore, a well-indexed stopword list is necessary, as there is currently no standardized stopword lexicon [1].

Analysis of sentiment on the Indonesian language Since YouTube is one of the primary channels for Indonesians to voice their opinions, YouTube comments are significant. With millions of followers, channels like MiawAug reflect public opinion in addition to providing a forum for amusement. A more thorough comprehension of sentiment on this platform can help advertisers, content producers, and even the government understand how the public feels about particular topics or goods that are advertised on YouTube.

It is hoped that the results of this study will provide a deeper understanding of the advantages and disadvantages of each algorithm in the context of sentiment analysis. These findings are also expected to assist researchers and practitioners in selecting the best algorithm for sentiment analysis on other social media platforms.

II. RESEARCH METHODOLOGY

Figure 1 illustrates the research stages that follow the general process in sentiment analysis, using the slang word conversion approach.

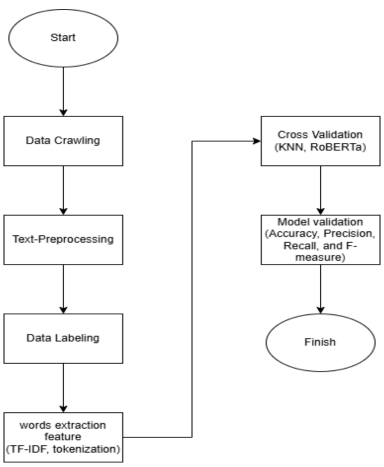


Figure 1. Research Flow

The flowchart illustrates the steps in a machine learning pipeline for text classification, specifically in sentiment analysis, using K-Nearest Neighbors (KNN) and RoBERTa.

A. Data Crawling

This research dataset consists of opinionated comment text written in Indonesian. Information is obtained from comments from the MiawAug Youtube channel by crawling. Data was taken from the video comment column uploaded in April 2024 with as many as 7,619 comments, which can be seen in Table 1. Crawling uses an algorithm developed by researchers in the Python programming language.

Table 1. Data Source

No	Video URL	
1	https://www.youtube.com/watch?v=MP4rX9ClP1Q *Serem Banget Ini Game, Nggak Boleh Teriak Juga !!!* - Silent Breath Indonesia	
	Upload Date	April 9, 2024
	Viewer	1,956,563
	Number of comments	2,826
	Date of Crawling by researchers	September 1, 2024
2	https://www.youtube.com/watch?v=xdi4mjBfd94&t=7s *Pacar Yang Menyeramkan* - The Stalked Horror Game Indonesia	
	Upload Date	April 2, 2024
	Viewer	1,880,419
	Number of comments	2,852
	Date of Crawling by researchers	September 1, 2024
3	https://www.youtube.com/watch?v=8U9B90I696U&t=13s *Tersesat Di Desa Berhantu* - I'm Counting To 6 Indonesia	
	Upload Date	April 16, 2024
	Viewer	1,266,889
	Number of comments	1,941
	Date of Crawling by researchers	September 1, 2024

Every comment on all three videos will be fully utilized in this research. This allows us to see how many people have viewed MiawAug's YouTube channel.

B. Text Preprocessing

Text preprocessing is the process of getting clean data ready for use in research [8]. Various preprocessing procedures were performed, including:

1. Data Cleaning

At this stage any symbols, punctuation marks, and other characters will be removed, leaving only phrases without these elements remaining after the cleaning process [9].

Table 2. Cleansing Data

Before	After
Minta dong link game nya...😭😭😭😭	*Minta dong link game nya*

In Table 2 above, it can be seen that the emoji has been removed.

2. Case Folding

After completing the cleaning step, the data or comment will go through a case folding process, which converts capital letters into lowercase letters, resulting in a comment that consists entirely of lowercase letters [9].

Table 3. Case Folding

Before	After
Minta dong link game nya	*minta dong link game nya*

The term *Minta*, which began with capital letters, has been modified to *minta* with all lowercase characters, as seen in Table 3 above.

3. Normalization

At this point, comments containing ambiguous or incomplete language will be converted into more precise language [9].

Table 4. Normalization

Before	After
yg di wahana aku taget bagetttt	*yang wahana kaget banget*

In order to comply with proper and decent Indonesian writing, the words *yg, taget,* and *bangettt* have been changed to the words *yang, kaget,* and *banget,* respectively, in Table 4 above.

4. Stemming

At this stage, all words that have affixes will be removed [9].

Table 5. Stemming

Before	After
grafiknya kek beneran real	*grafik kek beneran real*

To make the text more succinct, the word *grafiknya* has been replaced with *garfik* in Table 5 above.

5. Tokenization

The fifth step in the preprocessing procedure is this step. At this point, each word will be reduced to a single word. This is done so that the RoBERTa algorithm can process the text.

Table 6. Tokenization

Before	After
takut game	*['takut', 'game']*

The term *takut game* is transformed into a list format in Table 6 with each word as a distinct element: [*'takut', 'game'*]. This list can subsequently be processed further or input into a machine learning model.

6. TF - IDF weighting

TF - IDF weighting is the process of assigning weights to each word to maximize the performance of sentiment analysis. While IDF (Inverse Document Frequency) is a token weighting technique that tracks the occurrence of a token in a text set, TF (Term Frequency) can balance the importance based on the overall occurrence of the text in the text set [10].

7. Visualization

This visualization takes the form of a wordcloud, which is a graphical depiction of the most frequently occurring terms in the data. The text will appear at varying sizes, with the size of the words increasing according to their frequency of occurrence, getting larger when the data shows a higher frequency of occurrence [11].

8. Confusion Matrix

The last step involves assessing the previously processed phases. It is now important to be able to evaluate the accuracy level of the previous approach. Measurement using the Confusion Matrix approach, which has four characteristics—True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)—is required to obtain assessment findings. The performance of a matrix is evaluated using f1-score, recall, accuracy, and precision [12]. An illustration of how to determine the evaluation values is given below:

a) Accuracy

Model accuracy refers to how well the model classifies data (Equation 1).

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

The number of cases that were accurately anticipated to be positive is known as True Positives (TP). The number of cases that were accurately anticipated to be negative is known as True Negatives (TN). The number of cases that were mispredicted as positive is called False Positives (FP), or Type I error. The number of cases that were mispredicted as negative is known as False Negatives (FN), or Type II error.

The accuracy metric calculates the proportion of correct predictions (both true positives and true negatives) to the total number of predictions made (sum of true positives, true negatives, false positives, and false negatives). It gives an overall measure of how well the model is performing across all classes.

In other words, accuracy reflects the ratio of correct predictions to the total predictions made by the model [6], [12].

b) Precision

Precision is the level of accuracy between the model's prediction results and the requested data (Equation 2).

$$precision = \frac{TP}{TP+FP} \quad (3)$$

Where, true Positives (TP) are the number of instances correctly predicted as positive, while False Positives (FP) are the number of instances incorrectly predicted as positive.

Precision measures the accuracy of positive predictions made by the model. It represents the proportion of true positive predictions out of all instances predicted as positive. In other words, precision tells us how many of the model's positive predictions were actually correct.

In situations when the cost of false positives is significant, like spam detection (where we want to avoid labeling real emails as spam), precision is very helpful [6], [12].

c) Recall

Recall or sensitivity is the term used to describe the model's capacity to successfully retrieve information (Equation 3).

$$recall = \frac{TP}{TP+FN} = \frac{TN}{P} \tag{3}$$

The formula provides two methods for calculating recall. The ratio of true positives (TP) to total true positives (TP) plus false negatives (FN) is known as recall. Another way to compute recall is to divide the number of positives (P) by the true negative (TN).

The model's recall quantifies its capacity to recognize every accurate positive example. A high recall value means that all of the right positive instances were correctly identified by the model [6], [12].

d) F1-Score

The weighted average comparison between precision and recall is known as F1-Score (Equation 4). If the proportion of false positive and false negative data in our dataset is symmetrical, then accuracy can be used as a benchmark for algorithm performance. However, we should use F1-Score as a guide if the number is not close to [6], [12].

$$F1 = \frac{2.(Recall.Precision)}{Recall+Precision} \tag{4}$$

According to the formula, the F1 score is determined by taking the product of precision and memory, multiplying it by 2, and then dividing the result by the total of precision and recall.

C. Data Labeling

The selected opinion sentences are labeled. Opinion labels consist of neutral, positive and negative sentiments. The labeling process is done automatically by using the pattern library. So there is no need to do labeling manually. An example of purchasing labels is in Table 7.

Table 7. Labeling Data

No	Text	Label
1	*grafik real banget keren*	positive
2	*lucu*	neutral
3	*takut game*	negative

D. Data Resampling

Data resampling is a strategy that involves changing the sample size or distribution of a dataset. Resampling is often used to solve class imbalance problems, improve model generalization, or reduce bias and uncertainty. In this study, researchers used the Oversampling method where the minority class will be added to the sample data so that it can be balanced with the majority class [13].

E. Modeling

1. K-Fold Cross Validation

It is a method to separate data into test sets and training sets. Furthermore, the separation of data into k sets of subdata in equal amounts is implemented by K-Fold Cross Validation [14]. In this case, K-Fold is performed five times with a ratio of 80% training data and 20% test data.

2. A Robustly Optimized BERT Pretraining Approach (RoBERTa)

RoBERTa (Robustly Optimized BERT Pretraining Approach) is an improved and optimized BERT pre-training model developed by Facebook AI Research (FAIR) [5], [7].

3. K-Nearest Neighbour (KNN)

The K-Nearest Neighbor (KNN) method uses the closest distance to determine its classification (Equation 5) [10], [13]. The Euclidean distance between the test sample and the specified training sample is usually the basis of the KNearest Neighbor classifier. The k neighbors that are closest to you are listed below.

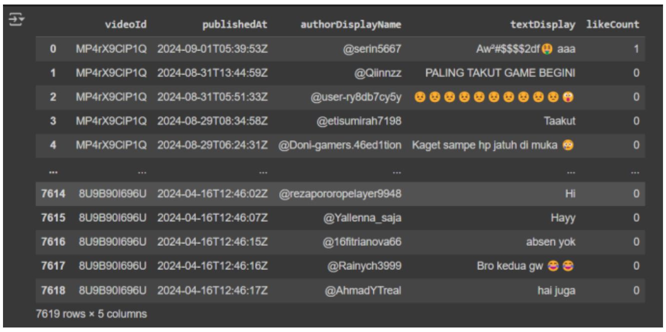
$$D(p,q) = \sqrt{\sum_i^n (q_i - p_i)} \tag{5}$$

In the above formula, qi is the data with normalized properties, and Pi is the new test data overlaid on top of the training data.

III. RESULTS AND DISCUSSION

A. Data Crawling Results

The results of Crawling conducted on September 1, 2024, can be seen in Figure 2 below, which amounts to 7,619 raw data.



	videoId	publishedAt	authorDisplayName	textDisplay	likeCount
0	MP4rX9CIP1Q	2024-09-01T05:39:53Z	@serin5667	Aw²#$$$$2df🤑 aaa	1
1	MP4rX9CIP1Q	2024-08-31T13:44:59Z	@Qiinnzz	PALING TAKUT GAME BEGINI	0
2	MP4rX9CIP1Q	2024-08-31T05:51:33Z	@user-ry8db7cy5y	😟😟😟😟😟😟😟😟😟😟😲	0
3	MP4rX9CIP1Q	2024-08-29T08:34:58Z	@etisumirah7198	Taakut	0
4	MP4rX9CIP1Q	2024-08-29T06:24:31Z	@Doni-gamers.46ed1tion	Kaget sampe hp jatuh di muka 😳	0
...	...	...	...	...	...
7614	8U9B90I696U	2024-04-16T12:46:02Z	@rezapororopelayer9948	Hi	0
7615	8U9B90I696U	2024-04-16T12:46:07Z	@Yallenna_saja	Hayy	0
7616	8U9B90I696U	2024-04-16T12:46:15Z	@16fitrianova66	absen yok	0
7617	8U9B90I696U	2024-04-16T12:46:16Z	@Rainych3999	Bro kedua gw 😂😂	0
7618	8U9B90I696U	2024-04-16T12:46:17Z	@AhmadYTreal	hai juga	0

7619 rows × 5 columns

Figure 2. Data Crawling Results

DOI: 10.34148/teknika.v14i1.1117

The Figure 2 above contains information about videos on a platform. The columns are:

- videoId : The unique ID of the video
- publishedAt : Date and time the video was published
- authorDisplayName : Username of the person who uploaded the video
- textDisplay : The text displayed in the video
- likeCount : The number of likes the video received

The Figure 2 shows a list of videos with the name of the uploader, the date of publication, the text displayed in the video, and the number of likes it received.

B. Text Pre-processing Result

The pre-processing result of one sample text is shown in Table 8. Starting with the raw data obtained through crawling results, the procedure proceeds through several steps.

Table 8. Text Preprocessing Result

Steps	Result
Raw Data	*Apaan sih random banget setannya Dateng" langsung ninju* 😂 .. *sakit perut ketawa ..*
Data Cleaning and Case Folding	*apaan sih random banget setannya datengquot langsung ninju sakit perut ketawa*
Stemming	*apa sih random banget setan datengquot langsung ninju sakit perut ketawa*
Normalization	*random banget setan datang langsung ninju sakit perut ketawa*
Tokenization	*['random', 'banget', 'setan', 'datang', 'langsung', 'ninju', 'sakit', 'perut', 'ketawa']*

C. Wordcloud

Wordcloud is a graphical depiction of text that highlights words according to their frequency of occurrence. The size of words that appear most frequently in the text will be larger, while the size of words that appear less frequently will be smaller [11]. This can be seen from Figures 3, 4, and 5 below.

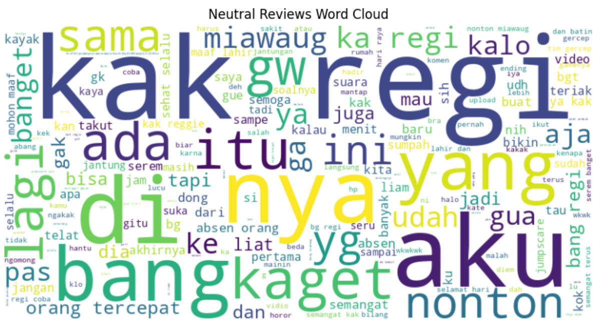


Figure 3. Wordcloud Neutral

The image is a word cloud of Indonesian words that are associated with neutral reviews. The largest words are *"sama"*, *"kak"*, *"ada"*, *"itu"*, *"ini"*, *"yang"*, *"aku"*, and *"nonton"*. These words are all common in Indonesian language and they suggest that the reviews were neither very positive nor very negative. The smaller words give further details about the specific aspects of the product or service that were neutral.

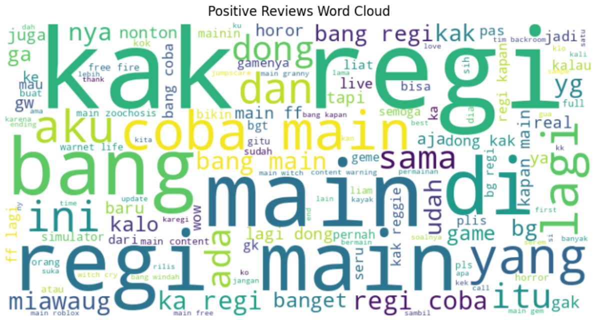


Figure 4. Wordcloud Positive

This is a word cloud of positive reviews. Some of the most common words are "main", "bang", "kak", "regi", "cobal" and "game". There are also words that suggest positive emotions like "suka" (like) and "seru" (fun). These words suggest that the reviews were very positive.

Figure 5. Wordcloud Negative

The word cloud is a visualization of the most frequent words used in negative reviews of a game. The largest words in the cloud are "game", "kak", "lagi", "main", and "horor". This suggests that many of the negative reviews mentioned the game itself, and also complained about things like "horror", "kaget" (scared), and "bang" (loud noise), which are common problems with horror games. There are also words related to bugs and gameplay, like "reggi" (a type of bug), "update", and "teriak" (screaming), which are common in negative reviews of any game.

D. Labeling Result

In the labeling process using the library pattern, it produces data in the form of neutral labels totaling 5,143, positive labels totaling 718, and negative labels totaling 1,073, which can be seen in Figure 6 below.

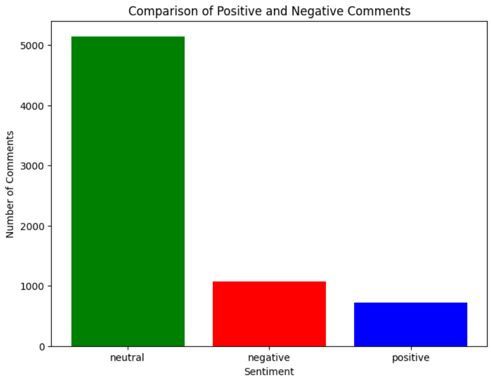


Figure 6. Labeling Result

The figure above shows the sentiment distribution in the comment dataset. It shows that the majority of comments are neutral (green), followed by negative comments (red) and then positive comments (blue). This shows that the overall sentiment towards the subject of these comments is not overwhelmingly positive or negative.

E. Resampling Result

Resampling of data is done to balance the labeled data due to unbalanced data. Oversampling is the technique used, where the sample size of the minority class is enlarged to equal that of the majority class.

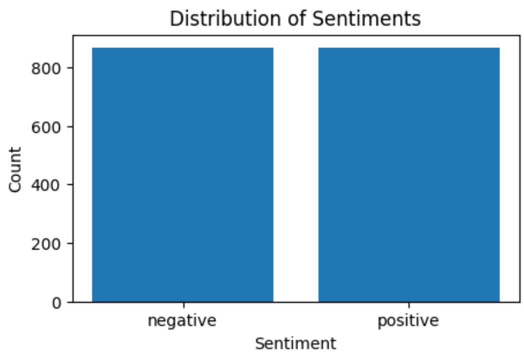


Figure 7. Resampling 2 Class

The data for the positive and negative classes has been balanced, as seen in Figure 7 above.

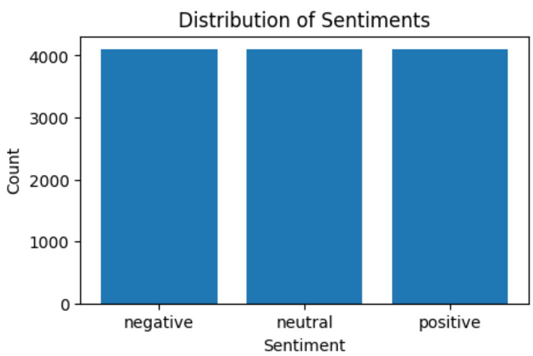


Figure 8. Resampling 3 Class

The data for the neutral, negative, and positive classifications are balanced, as shown in Figure 8 above.

F. Sentiment Comparison Results of KNN and RoBERTa Algorithms

In this study, a number of test scenarios were conducted using two class variations: two-class (positive and negative) and three-class (neutral, positive, and negative) scenarios. The data used by both algorithms was identical, with 20% used for testing and 80% for training. Furthermore, as shown in the table and figure below, both algorithms used k-fold cross validation with five folds for each scenario.

Table 9. KNN 2 Class Testing Results

Fold	Accuracy	Precision	Recall	F1 Score
1	84.20 %	77.52 %	96.57 %	86.01 %
2	86.74 %	79.70 %	96.99 %	87.50 %
3	83.57 %	75.73 %	95.71 %	84.55 %
4	90.20 %	85.64 %	96.53 %	90.76 %
5	85.88 %	80.87 %	97.38 %	88.36 %

Based on the Table 9 above, the KNN algorithm achieved the following results for the 2-class category: accuracy of 86.12 ± 2.34%, F1-Score of 87.44 ± 2.11%, recall of 96.64 ± 0.56%, and precision of 79.89 ± 3.38%.

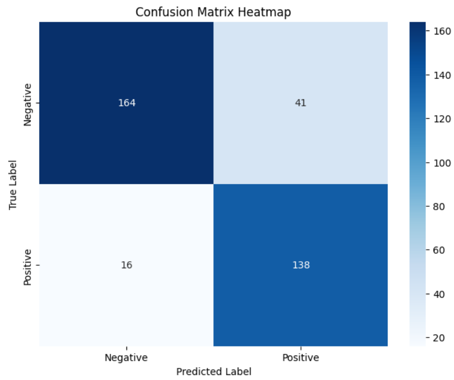


Figure 9. Confusion Matrix KNN 2 Class

The Figure 9 shows a confusion matrix, which is a table that summarizes the performance of a classification model. It shows the number of correct and incorrect predictions for each class.

The matrix is divided into four quadrants:

- **True Negative (TN):** This quadrant shows the number of negative instances that were correctly predicted as negative. In this case, 164.
- **False Positive (FP):** This quadrant shows the number of negative instances that were incorrectly predicted as positive. In this case, 41.
- **False Negative (FN):** This quadrant shows the number of positive instances that were incorrectly predicted as negative. In this case, 16.
- **True Positive (TP):** This quadrant shows the number of positive instances that were correctly predicted as positive. In this case, 138.

DOI: 10.34148/teknika.v14i1.1117
TEKNIKA, Volume 14(1), March 2025, pp. 26-33
ISSN 2549-8037, EISSN 2549-8045

The heatmap color scheme shows the relative number of occurrences in each cell. The darker the blue, the more occurrences.

This confusion matrix is a valuable tool for evaluating the performance of a model, particularly for understanding what types of errors the model is making.

Table 10. KNN 3 Class Testing Results

Fold	Accuracy	Precision	Recall	F1 Score
1	98.34 %	98.35 %	98.34 %	98.34 %
2	98.25 %	98.26 %	98.25 %	98.25 %
3	97.40 %	97.46 %	97.40 %	97.41 %
4	98.54 %	98.55 %	98.54 %	98.54 %
5	98.54 %	98.55 %	98.54 %	98.54 %

Based on the Table 10 above, the accuracy of the KNN algorithm, F1-Score, recall, and precision for the 3-class category were 98.21 ± 0.42%, 98.21 ± 0.42%, and 98.23 ± 0.40%, respectively.

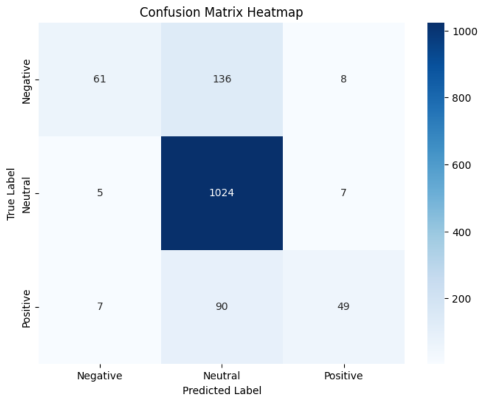


Figure 10. Confusion Matrix KNN 3 Class

A confusion matrix heatmap is displayed in Figure 10. The heatmap displays a categorization model's performance. The anticipated labels are shown in the columns, and the genuine labels are shown in the rows. The value in each cell indicates how many cases were classified correctly or incorrectly.

Table 11. RoBERTa 2 Class Testing Results

Fold	Accuracy	Precision	Recall	F1 Score
1	91.38 %	90.05 %	93.14 %	91.57 %
2	93.08 %	91.27 %	94.57 %	92.89 %
3	94.52 %	93.37 %	95.09 %	94.22 %
4	94.81 %	93.29 %	96.53 %	94.88 %
5	97.11 %	98.39 %	96.33 %	97.35 %

For the 2-class category, the RoBERTa algorithm produced an accuracy of 93.89 ± 2.20%, an F1-Score of 93.88 ± 2.25%, precision of 93.22 ± 3.06%, and recall of 94.59 ± 2.15% (Table 11).

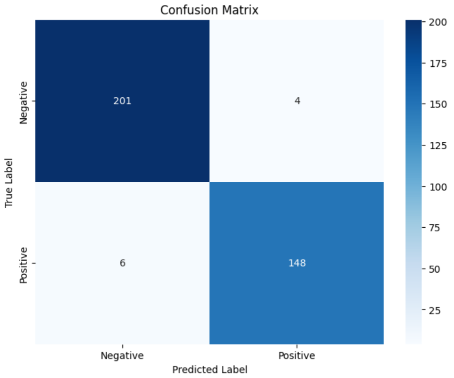


Figure 2. Confusion Matrix RoBERTa 2 Class

Figure 11 shows the confusion matrix for RoBERTa 2 Class classification task. The diagonal elements represent the number of correctly classified examples, while the off-diagonal elements represent the number of incorrectly classified examples.

- **True Negative (TN):** 201 - The model correctly predicted 201 negative examples.
- **False Positive (FP):** 4 - The model incorrectly predicted 4 negative examples as positive.
- **False Negative (FN):** 6 - The model incorrectly predicted 6 positive examples as negative.
- **True Positive (TP):** 148 - The model correctly predicted 148 positive examples.

The confusion matrix provides information about the performance of the model for different classes.

Table 12. RoBERTa 3 Class Testing Results

Fold	Accuracy	Precision	Recall	F1 Score
1	99.14 %	99.16 %	99.14 %	99.14 %
2	98.74 %	98.74 %	98.75 %	98.74 %
3	98.94 %	98.94 %	98.94 %	98.94 %
4	99.31 %	99.31 %	99.31 %	99.31 %
5	98.98 %	99.03 %	98.95 %	98.97 %

RoBERTa achieved an accuracy of 99.21 ± 0.25%, F1-Score 99.21 ± 0.25%, precision 99.21 ± 0.26%, and recall 99.21 ± 0.24% for the 3-class category (Table 12).

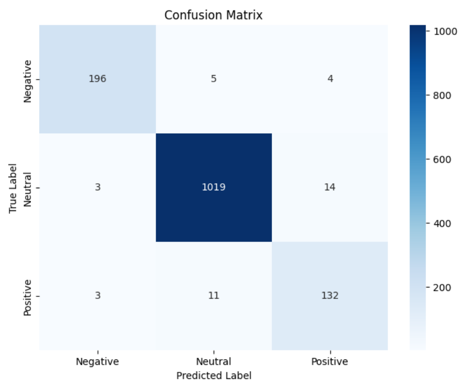


Figure 3. Confusion Matrix RoBERTa 3 Class

DOI: 10.34148/teknika.v14i1.1117
ISSN 2549-8037, EISSN 2549-8045

The confusion matrix (Figure 12) shows the results of the RoBERTa 3 Class classification model. The rows represent the actual classes (true labels), and the columns represent the predicted classes. The numbers in each cell represent the number of instances that were classified correctly or incorrectly. For example, the cell in the second row and second column shows that 1019 instances were correctly classified as neutral. The confusion matrix can be used to evaluate the performance of a classification model. For example, it can be used to calculate the accuracy, precision, recall, and F1-score.

Based on this confusion matrix, the model seems to be performing well. It is particularly good at classifying instances as neutral. However, it does struggle with classifying instances as positive.

IV. CONCLUSION

Based on the above research, the RoBERTa algorithm outperforms the K-Nearest Neighbor (KNN) method in sentiment analysis of comments on MiawAug's YouTube channel. A total of 7,619 raw data were obtained from the crawling operation on September 1, 2024, and the pattern library was used for labeling. Based on the labeling results, there are 1,073 negative comments, 718 positive comments, and 5,143 comments with neutral labels.

KNN was only able to achieve 86.12% accuracy in the 2-class category, but RoBERTa recorded 93.89% accuracy, 93.88% F1-Score, 94.59% recall, and 93.22% precision. With 99.21% accuracy, 99.21% F1-Score, 99.21% recall, and 99.21% precision, RoBERTa outperformed KNN in the 3-class category, while KNN obtained 98.21% accuracy. Therefore, RoBERTa has better performance, especially in the 3-class category, in terms of accuracy, F1-Score, recall, and precision. Based on these findings, RoBERTa is recommended over KNN for more complicated multiclass sentiment analysis.

REFERENCES

[1] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, “Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19),” *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 406, 2021, doi: 10.30865/mib.v5i2.2835.

[2] N. Putu *et al.*, “Analisa Penerapan Tax Treaty Indonesia Dengan Amerika Dalam Pajak Adsense Youtube Partner Program Di Indonesia,” *J. Kertha Desa*, vol. 11, no. 6, pp. 2704–2714, 2023, [Online]. Available: https://voi.id/teknologi/39265/youtuber-kena-pajak-pemerintah-as-begini-hitung-

[3] M. Takhim, A. I. Fadila, and M. Maskudi, “Monetasi Youtube Perspektif Fikih Muamalah,” *J. Multidisiplin Madani*, vol. 2, no. 2, pp. 1019–1034, 2022, doi: 10.54259/mudima.v2i2.484.

[4] A. S. Lubis, “Effect of Adsense on Deddy Corbuzier ’ s YouTube on the Viewing Interests and Viewing Actions of Universitas Sumatera Utara Students,” no. July 2006, pp. 666–671, 2023.

[5] S. Informasi, F. Teknik, and U. Bakrie, “Perbandingan Performa Algoritma VADER dan RoBERTa pada Twitter Comparison of VADER and RoBERTa Algorithm Performance on,” vol. 13, pp. 1547–1557, 2024.

[6] J. W. Iskandar and Y. Nataliani, “Perbandingan Naïve Bayes, SVM, dan k-NN untuk Analisis Sentimen Gadget Berbasis Aspek,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1120–1126, 2021, doi: 10.29207/resti.v5i6.3588.

[7] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” no. 1, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

[8] K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, “Sentiment Analysis With Ensemble Hybrid Deep Learning Model,” *IEEE Access*, vol. 10, no. September, pp. 103694–103704, 2022, doi: 10.1109/ACCESS.2022.3210182.

[9] A. A. Azhari, Y. Sibaroni, and S. S. Prasetiyowati, “Detection of Indonesian Hate Speech in the Comments Column of Indone-sian Artists’ Instagram Using the RoBERTa Method,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 8, no. 3, pp. 764–773, 2023, doi: 10.29100/jipi.v8i3.3898.

[10] M. J. Palepa, N. Pratiwi, and R. Q. Rohmansa, “Analisis Sentimen Masyarakat Tentang Pengaruh Politik Identitas Pada Pemilu 2024 Terhadap Toleransi Beragama Menggunakan Metode K - Nearest Neighbor,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 1, pp. 389–401, 2024, doi: 10.29100/jipi.v9i1.4957.

[11] G. R. Putri, M. A. Maulana, and S. Bahri, “Perbandingan Algoritma Naïve Bayes dan TextBlob Untuk Mendapatkan Analisis Sentimen Masyarakat Pada Sosial Media,” *Teknika*, vol. 13, no. 2, pp. 213–218, 2024, doi: 10.34148/teknika.v13i2.815.

[12] M. A. A. Yani and W. Maharani, “Analyzing Cyberbullying Negative Content on Twitter Social Media with the RoBERTa Method,” *JINAV J. Inf. Vis.*, vol. 4, no. 1, pp. 61–69, 2023, doi: 10.35877/454ri.jinav1543.

[13] Athirah Rifdha Aryani and Erwin Budi Setiawan, “Big Five Personality Assessment Using KNN method with RoBERTA,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 5, pp. 818–823, 2022, doi: 10.29207/resti.v6i5.4394.

[14] R. A. Putra, R. Novita, T. K. Ahsyar, and Zarnelly, “Implementation of Classification Algorithm for Sentiment Analysis: Measuring App User Satisfaction,” *Teknika*, vol. 13, no. 2, pp. 204–212, 2024, doi: 10.34148/teknika.v13i2.827.