International Journal Of Health, Engineering And Technology (IJHET) *E-ISSN 2829 - 8683*
Volume 4, Number 6, March 2026, *Page. 2732 - 2743*
Email : editorijhess@gmail.com

Public Sentiment Analysis of the Indonesian National Team's Performance Under Patrick Kluivert Using the Random Forest Algorithm

M Rifqi Tri Putra[1)], Rendra Gustriansyah[2)], Lastri Widya Astuti[3)]
[1,2,3)]Informatics Engineering Study Program, Faculty of Computer Science & Science, Indo Global Mandiri University

*Corresponding Author
Email : 021110052@students.uigm.ac.id

Abstract

This study aims to analyze public sentiment towards the performance of the Indonesian National Football Team during the Patrick Kluivert era using the Random Forest Classifier algorithm. The research data was obtained from 1,000 Twitter tweets collected through web scraping with relevant keywords, between January and March 2025. The obtained data was processed through the stages of cleaning, case folding, tokenization, stemming, and stopword removal, then converted into numeric form using the Term Frequency–Inverse Document Frequency (TF-IDF) method. Sentiment was categorized into two classes, namely positive and negative. The test results showed that the Random Forest model was able to classify sentiment with an accuracy rate of 83% with a precision value of 100%, a recall of 33.33%, and an F1-score of 50%. This finding confirms that public opinion towards the Indonesian National Team during the Kluivert era is divided into two main tendencies: positive support and negative criticism. This study proves that the Random Forest algorithm is effective for social media-based sentiment analysis and can be a reference for the PSSI and related parties in understanding public perception to improve the quality of team strategy and performance.

Keywords: Analysis, Public, Performance, Football, Indonesia

INTRODUCTION

Football is the most popular sport in Indonesia, serving not only as entertainment but also as a means of national unity. The high level of public interest in this sport is evident in the intense discussions, from children to adults. However, despite such strong public support, the Indonesian National Team (Timnas), especially the senior team, still faces difficulties achieving success in Asian and international tournaments. Since independence, the Indonesian National Team has never qualified for the World Cup, and its best opportunity is usually only in the AFF Cup.[1] The change of coach from Shin Tae-yong to Patrick Kluivert in January 2025 marked an effort by the PSSI (Indonesian Football Association) to adopt the Dutch football philosophy and utilize the potential of diaspora players. Kluivert, a former Dutch national team striker with experience coaching various clubs, was expected to bring a more modern strategy. However, this change sparked pros and cons among the public, reflected in opinions on social media.

The digital era has made social media, particularly Twitter, a primary means for people to express their views. According to data from Social (2022), there are 191.4 million social media users in Indonesia, representing 88.5% of the population aged 13 and over, and Twitter is among the top five most-used platforms. This large user base presents an opportunity to extract public opinion in real time through sentiment analysis. Twitter provides a rich and diverse data source on national football topics, making it relevant for research.

Several previous studies have utilized machine learning algorithms to analyze public sentiment. For example, Twitter sentiment analysis using Random Forest on hotel customers in Purwokerto achieved 87.23% accuracy . Sentiment analysis related to online learning policies using K-NN achieved 84.93% accuracy. Meanwhile, sentiment analysis of the Indonesian National Team after the 2020 AFF Cup final showed different performance between the K-NN and Random Forest algorithms [4]. However, a study specifically highlighting the performance of the Indonesian National Team during the Patrick Kluivert era has never been conducted before, so this research fills that gap.

Based on the background and brief literature review, this study aims to analyze public

sentiment toward the performance of the Indonesian National Team during Patrick Kluivert's coaching tenure using the Random Forest algorithm. This method was chosen because of its ability to combine predictions from various decision trees, resulting in a more stable and accurate classification than a single approach. The results are expected to provide in-depth insight into public opinion trends (positive and negative) and provide input for the Football Association of Indonesia (PSSI), coaches, and sports media in formulating communication strategies and improving the performance of the Indonesian National Team on the international stage.

RESEARCH METHODS

This study uses a quantitative text mining approach to analyze public sentiment toward the performance of the Indonesian National Football Team during the Patrick Kluivert era. Data was collected from Twitter using web scraping techniques with relevant keywords from January to March 2025, resulting in 1,000 tweets serving as the research dataset. Irrelevant public comments were eliminated using a purposive sampling approach to ensure the analyzed data was more focused and representative of the research topic.

The obtained data is then processed through pre-processing stages which include cleaning to remove punctuation, symbols, URLs, hashtags, and usernames, then case folding to standardize letters to lowercase, followed by tokenization to break the text into words, stemming to convert affixed words to their basic form, and stopword removal to filter out words that have no significant meaning. After this stage, each tweet is manually labeled with positive or negative sentiment with the help of the Python NLTK and Sastrawi libraries to form training and test data ready to be used for model training.

The labeled text data was then converted into a weighted numeric representation using the Term Frequency–Inverse Document Frequency (TF-IDF) method to measure the importance of words in each tweet relative to the entire corpus. The resulting TF-IDF representation was used as input for the classification process using the Random Forest algorithm. This algorithm was chosen for its ability to combine multiple decision trees to improve prediction stability and accuracy and reduce the risk of overfitting.

Model performance was evaluated using accuracy, precision, recall, and F1-score metrics to provide a comprehensive overview of the model's ability to accurately classify positive and negative sentiment. This comprehensive approach was designed to provide accurate classification results and benefit stakeholders in understanding public perception of the Indonesian national team under Patrick Kluivert's coaching.

RESULTS AND DISCUSSION

This study successfully collected 1,000 reviews or public opinion data regarding the performance of the Indonesian National Football Team during the Patrick Kluivert era, obtained from social media platforms like Twitter. The dataset contained diverse comments, ranging from support and criticism to suggestions, which were then processed using text mining. After pre-processing processes such as cleaning, case folding, tokenization, stemming, and stopword removal, the data, originally in raw text form, was converted into a weighted numeric representation using the TF-IDF method.

Table 1. Dataset

No	Data
1	miss Shin Tae Yong
2	Just one word fire
3	Garuda Spirit
4	**The hope of getting through to the general election has been destroyed because of ego**
5	In firing STY Salute to coach PK and the coaching staff for giving more confidence **to th players who performed well in the previous match**

Data Preprocessing

The data pre-processing stage is carried out through five sequential processes: cleaning, case folding, tokenization, stemming, and stopword removal. Each stage is designed to optimize data quality before feature extraction is carried out.

Cleaning

Using a dataset of 1,000 reviews, the initial preprocessing step involved data cleaning. The goal of this process was to remove irrelevant characters, such as punctuation and special symbols. Figure 1 shows the code implementation for the cleaning process, while Table 2 presents an example of the cleaning results.

```
# Mendefinisikan fungsi clean_text
def clean_text(text):
    # Mengecek apakah text adalah string dan bukan NaN
    if isinstance(text, str):
        # Menghapus karakter khusus, angka, dan tanda baca
        text = re.sub(r'[^a-zA-Z\s]', '', text)

        # Menghapus emoji tertentu
        text = re.sub(r'🤗|😍|❤️|🔥|😇', '', text)
    else:
        # Jika bukan string (misalnya NaN atau angka), ganti dengan string kosong
        text = ''
    return text
```

Figure 1. Input Cleaning

Table 2. Cleaning Process Results

No	**Original Tweet**	**After Cleaning**
1	miss Shin Tae Yong	miss Shin Tae Yong
2	just one word fire	just one word fire
3	Garuda Spirit	Garuda Spirit
4	The hope of getting through to the general election has been destroyed because ego fires STY	The hope of getting through to the general election has been destroyed because ego fires STY
5	Salute to coach PK and the coaching staff for giving more confidence to players who performed well in the match.	Salute to coach PK and the coaching staff for giving more confidence to players who performed well in the match.

	previously	previously

Folding Case

```
def case_folding(text):
    # Memeriksa apakah teks adalah string
    if isinstance(text, str):
        # Mengubah teks menjadi lowercase
        text = text.lower()
    else:
        # Jika bukan string, bisa mengembalikan string kosong atau teks lainnya
        text = ''

    return text
```

Case folding is the process of converting all letters to lowercase. This process converts the characters 'A'-'Z' in the data into the characters 'a'-'z'. Figure 2 shows the code implementation for the case folding process, while Table 3 presents an example of the case folding results.

Figure 2. Input Case Folding

Table 3. Case Folding Process Results

No	**After Cleaning**	**After Case Folding**
1	miss Shin Tae Yong	I miss Shin Tae Yong
2	just one word fire	just one word fire
3	Garuda Spirit	Garuda spirit
4	The hope of getting through to the general election has been destroyed because ego fires STY	hopes of getting through to the general election have been destroyed because ego mecat sty
5	Salute to coach PK and the coaching staff for giving more confidence to players who performed well in the match. previously	Salute to coach pk and the coaching staff for giving more confidence to players who performed well in the match previously

Tokenize

At this stage, each word in a sentence in the document is separated. Word separation is generally done using spaces. While writing styles can vary, the main goal is to break the sentence into its constituent words. Figure 3 shows the code implementation for the tokenization process, while Table 4 presents an example of the tokenization results.

```
# kode untuk tokenizing
from nltk.tokenize import RegexpTokenizer

regexp = RegexpTokenizer (r'\w+|$[0-9]+|\S+')
sample['Token'] = sample['ulasan'].apply(regexp.tokenize)

# Save the DataFrame to an Excel file
output_file_path = 'tokenizing.xlsx'
sample.to_excel(output_file_path, index=False)
print(f"DataFrame berhasil disimpan ke file Excel: {output_file_path}")
```

Figure 3. Tokenize Input

Table 4. Tokenization Process Results

No	After Case Folding	After Tokenize
1	I miss Shin Tae Yong	['Miss', 'shin', 'tae', 'yong']
2	just one word fire	['only', 'one', 'word', 'fire']
3	Garuda spirit	['Spirit','garuda']
4	Hopes of getting through to the general election have been destroyed because of ego, which caused Sty to fire	['destroyed', 'already', 'hope', 'passed', 'to', 'pildun', 'because', 'ego', 'fire', 'sty']
5	Salute to Coach PK and the coaching staff for giving more confidence to the players who performed well in the previous match	['salute', 'same', 'coach', 'pk', 'and', 'line-up', 'coaching', 'give', 'trust', 'more', 'to', 'players', 'who', 'appear', 'good', 'in', 'match', 'previously']

Stemming

*Stemming*Converting words in a document to their base or root form. The stemming process in Indonesian documents is quite complex because it requires removing all affixes from the words. Figure 4 shows the code implementation for the stemming process, while Table 5 presents an example of the stemming results.

Figure 4. Input Stemming

Table 5. Stemming Process Results

No	After Tokenize	After Stemming
1	['miss', 'shin', 'tae', 'yong']	['miss', 'shin', 'tae', 'yong']
2	['only', 'one', 'word', 'fire']	['fire']
3	['spirit', 'garuda']	['spirit', 'garuda']
4	['destroyed', 'already', 'hope', 'passed', 'to', 'pildun', 'because', 'ego', 'fire', 'sty']	['destroyed', 'already', 'hope', 'passed', 'pildun', 'ego', 'fire', 'sty']
5	['salute', 'same', 'coach', 'pk', 'and', 'line', 'coaching', 'give', 'trust', 'more', 'to', 'players', 'who', 'appear', 'good', 'in', 'match', 'previous']	['salute', 'coach', 'pk', 'line up', 'train', 'believe', 'play', 'appear', 'good', 'match']

Stopword Remover

At this stage, a filtering process is carried out on words that are commonly used or rarely contribute significant meaning, known as stopwords. This process is called "Stopword Removal." By removing these words, this process can improve classification effectiveness, reduce data scatter, and

significantly reduce the dimensionality of the feature space. Figure 5 shows the code implementation for the stopword removal process, while Table 6 presents an example of the stopword removal results

Figure 5. Stopword Removel input

Table 6. Stopword Removal Process Results

No	After Stemming	After Stopword Removal
1	['miss', 'shin', 'tae', 'yong']	miss you, shin, tae, yong
2	['fire']	fired
3	['spirit', 'garuda']	spirit, Garuda
4	['destroyed', 'already', 'hope', 'passed', 'pildun', 'ego', 'mecat', 'sty']	destroyed, finished, hope, pass, pildun, ego, fired, sty
5	['salute', 'coach', 'pk', 'jajar', 'train', 'believe', 'play', 'perform', 'good', 'match']	salute, coach, pk, line up, train, believe, play, appear, good, compete

1. TF-IDF Labeling Process

This labeling process is carried out using the TF-IDF method. The TF-IDF weighting program code can be seen in Figure 6.

Figure 6. Tf-Idf weighting

Tf-Idf Weighting Results

```
{'miss': np.float64(0.5092), 'shin': np.float64(0.4710), 'tae':
np.float64(0.5092), 'yong': np.float64(0.5092)}

{'fire': np.float64(1.0)}

{'spirit': np.float64(0.7358), 'garuda': np.float64(0.6771)}
```

```
'ego': np.float64(0.4280), 'mecat': np.float64(0.4099), 'sty':
np.float64(0.2772)}

{'salut': np.float64(0.4163), 'coach': np.float64(0.2779), 'pk':
np.float64(0.3634), 'jajar': np.float64(0.3634), 'latih': np.float64(0.2181),
'percaya': np.float64(0.3529), 'main': np.float64(0.1656), 'tampil':
np.float64(0.3763), 'bagus': np.float64(0.2661), 'tanding':
np.float64(0.2705)}
```

After the word weighting process using the TF-IDF method is complete, the next step is to label the weighted results for classification purposes. This labeling aims to associate each text representation vector with the appropriate sentiment label, such as positive or negative, so that the data is ready for use in the model training process. The TF-IDF labeling process can be seen in Figure 7, which shows the relationship between feature values and their respective sentiment classes. This step is crucial in ensuring that the model can learn the correct patterns from the labeled data.

Figure 7. Tf-Idf labeling

Table 7 Tf-Idf Labeling Results

No	*Tf-Idf Values*	Sentiment
1	{'miss': np.float64(0.5092), 'shin': np.float64(0.4710), 'tae': np.float64(0.5092), 'yong': np.float64(0.5092)}	Positive
2	{'fire': np.float64(1.0)}	Positive
3	{'spirit': np.float64(0.7358), 'garuda': np.float64(0.6771)}	Positive
4	{'destroyed': np.float64(0.4280), 'already': np.float64(0.2578), 'hope': np.float64(0.3408), 'pass': np.float64(0.3228), 'pildun': np.float64(0.3189), 'ego': np.float64(0.4280), 'mecat': np.float64(0.4099), 'sty': np.float64(0.2772)}	Positive
5	{'salute': np.float64(0.4163), 'coach':	Negative

```
np.float64(0.2779), 'pk':
np.float64(0.3634), 'align': np.float64(0.3634), 'train':
np.float64(0.2181), 'believe': np.float64(0.3529),
'main':
np.float64(0.1656), 'appear': np.float64(0.3763), 'good':
np.float64(0.2661), 'match': np.float64(0.2705)}
```

Random Forest Implementation

The sentiment analysis model was built by integrating the Random Forest algorithm into the Google Colab platform. Pre-processed public opinion data was combined into strings for easy analysis, then separated into features (X) and labels.
(y). Next, the data is divided into training data and test data with a ratio of 80:20. Text features converted into a numerical representation using the TF-IDF Vectorizer so that the model can recognize the importance level of each word.

A Random Forest Classifier algorithm with 100 decision trees (n_estimators = 100, random_state = 42) was trained using TF-IDF transformed training data. The trained model was used to predict sentiment on unprocessed test data. Model performance was evaluated using a confusion matrix and accuracy, precision, recall, and F1-score metrics to assess the model's ability to accurately classify public sentiment.

Test results showed that Random Forest was able to classify negative sentiment with a zero error rate and achieved an overall accuracy of 83% in an 80:20 data split scenario. However, the model still showed weakness in the positive class, with relatively low recall values. Therefore, future training data balancing is needed to improve the performance of both classes.

Random Forest Testing

After the Random Forest model has been trained using the TF-IDF transformed training data, the next step is to test its performance on the test data. The evaluation is conducted using a confusion matrix and metrics such as accuracy, precision, recall, and F1-score. This approach provides a comprehensive overview of the model's ability to distinguish positive from negative sentiment.

The test results show that in the 80:20 data split scenario, the model achieved the highest accuracy of 83%, while in the 50:50 data split, the accuracy decreased to 77%. The confusion matrix shows that the model is very good at recognizing negative sentiment with a zero error rate (False Positive = 0) and 149 negative data were correctly classified (True Negative). However, there are still weaknesses in the positive sentiment because some positive data are predicted as negative (False Negative), so the recall value for the positive class is relatively low despite its high precision.

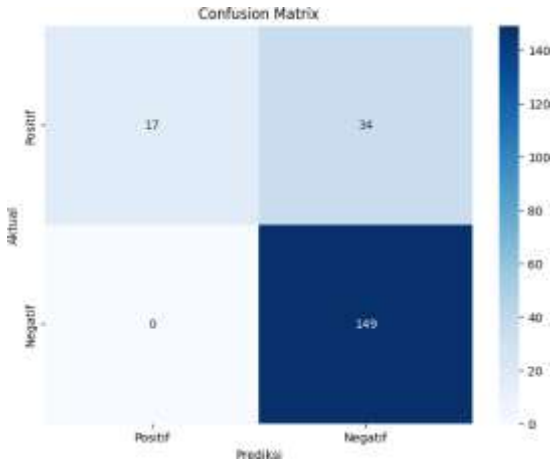

Figure 8. Confusion Matrix

Overall, the test results demonstrate that the Random Forest algorithm is effective for social media-based public sentiment analysis, but model performance is significantly affected by the balance

of training data and parameters used. These findings provide the basis for further development to allow the model to recognize both sentiment classes more effectively.

Decision Tree

The Random Forest algorithm works by constructing a number of decision trees, each trained using a randomly selected data sample and combination of features. The final prediction is obtained by aggregating the results of all the trees, for example by majority voting. This approach makes Random Forest more resistant to overfitting and remains accurate even when data is incomplete or imbalanced.

The decision tree visualization generated in this study demonstrates how the model distinguishes positive and negative sentiment based on the words used as features. The color and branching at each node depict the data distribution and the algorithm's chosen separation criteria, providing an interpretive overview of how the model makes decisions in classifying public sentiment, as shown in Figure 9.

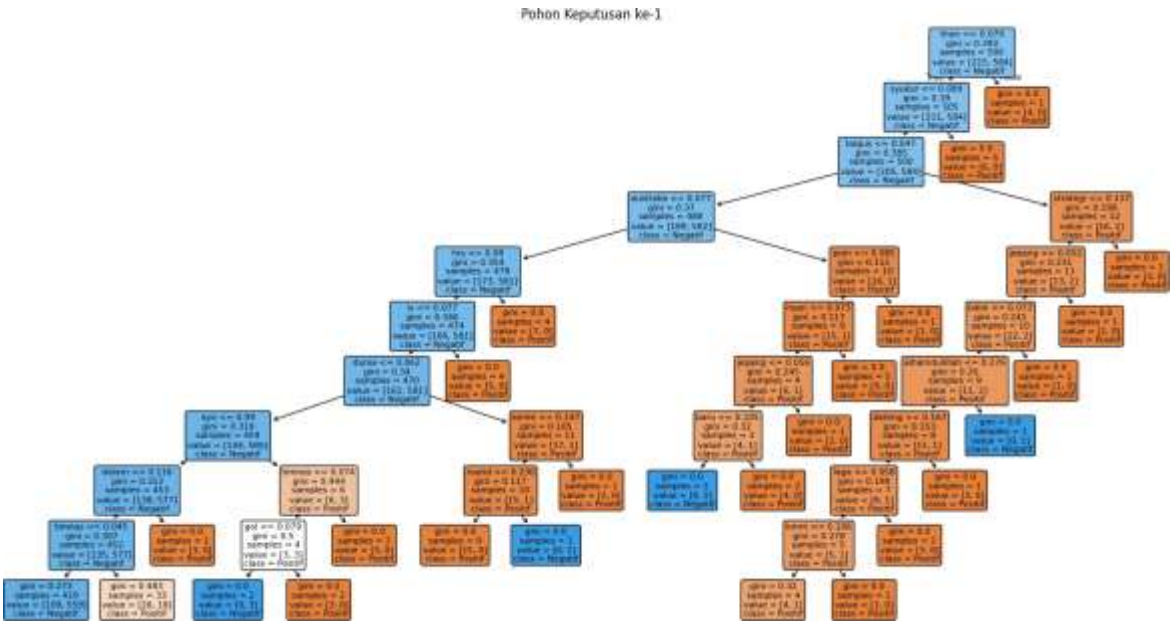

Figure 9. Example of Decision Tree Visualization of Random Forest Model Tree 1

The first decision tree structure visualization showing the branching of features to distinguish positive and negative sentiment.

A quality analysis of the five visualized trees shows that the first and fourth trees have simpler structures, many nodes with low Gini values (0.0), and clear branching, resulting in a cleaner and more balanced class differentiation. An example of the fourth tree can be seen in Figure 10.

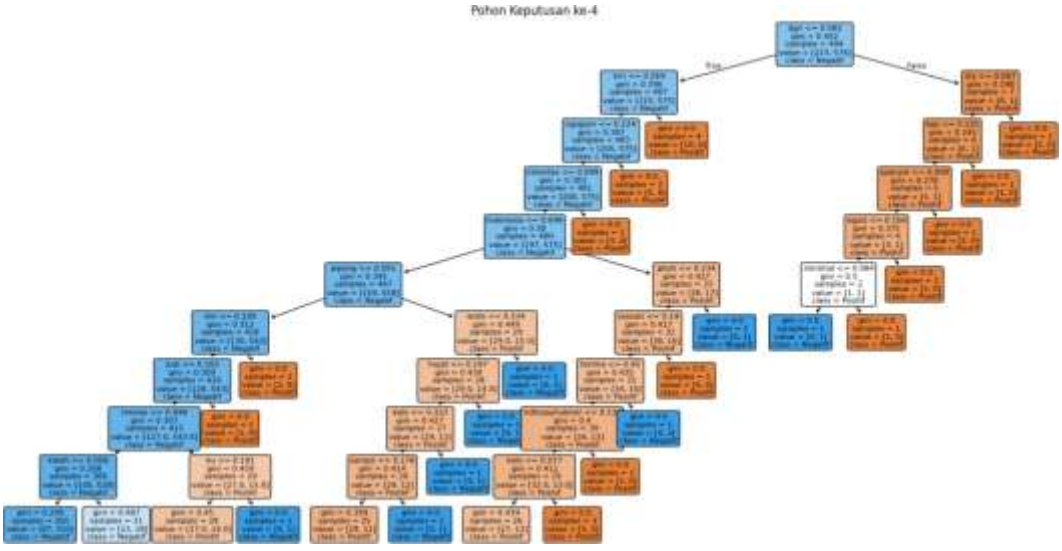

Figure 10. Example of Decision Tree Visualization of Random Forest Model Tree 4

In contrast, the fifth tree has the most complex structure, with many nodes with Gini values approaching 0.5 and long branching, making it prone to overfitting. An example of the fifth tree can be seen in Figure 11.

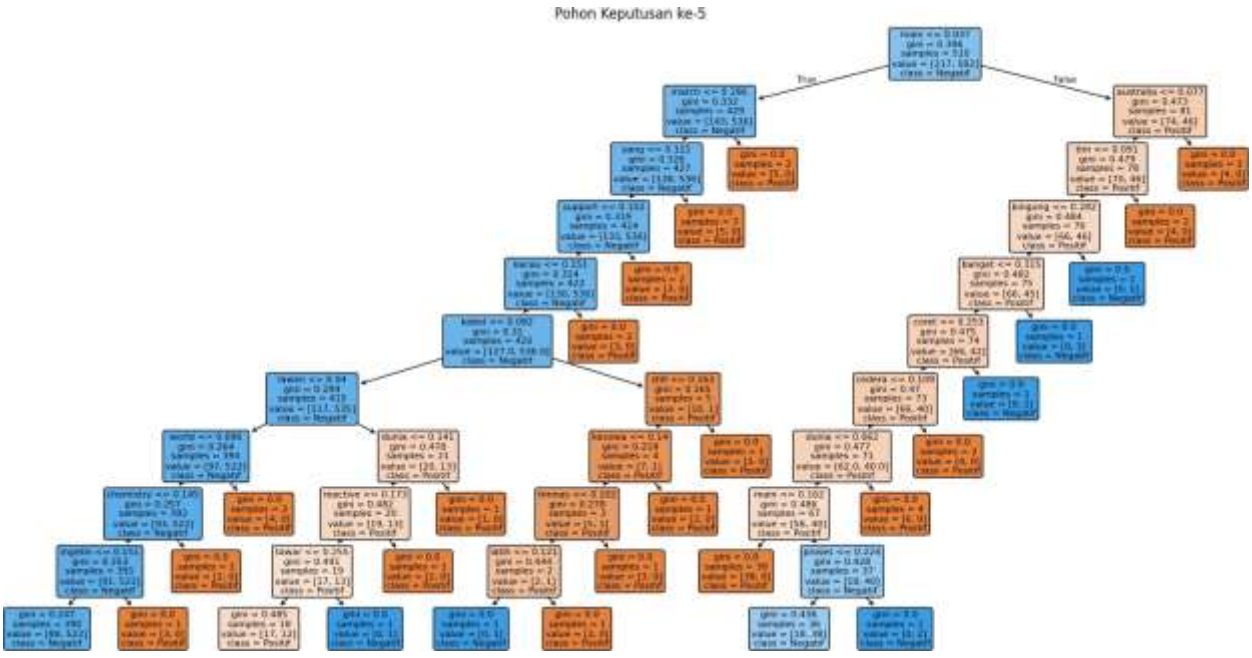

Figure 11. Example of Decision Tree Visualization of Random Forest Model Tree 5

The best nodes are found in all trees with a Gini value of 0.0, indicating perfect purity, for example, in the first tree with a node containing five positive samples. The worst nodes are found in branches with a Gini value close to 0.5 and a nearly balanced class distribution, such as in the second tree (value [12,11]), the third tree (value [13,11]), and the fifth tree (value [10,9]), indicating the model has difficulty distinguishing sentiment at that point.

These findings show that decision tree visualization is not only useful for model interpretation, but also for identifying the most effective and most error-prone parts of the model, which can serve as a basis for parameter improvement or future modeling strategies.

CONCLUSION

This study successfully analyzed public sentiment toward the performance of the Indonesian National Football Team during the Patrick Kluivert era using the Random Forest algorithm. Based on 1,000 Twitter tweets collected and processed through text mining, the Random Forest model achieved an accuracy of 83%, with high precision in the negative class but a relatively low recall value in the positive class. These results indicate that public opinion toward the Indonesian National Team is divided into two main tendencies: positive support and negative criticism. This study proves that the Random Forest algorithm is effective for social media-based sentiment analysis and can serve as a reference for the PSSI and related parties to understand public perception more deeply.

This study still has limitations in the data imbalance between positive and negative sentiment, which impacts model performance in certain classes. Therefore, future research is recommended to expand the data set to achieve a more balanced sentiment distribution and explore other machine learning algorithms or more complex ensemble methods to improve classification performance. Furthermore, the resulting decision tree visualization can be used as a basis for optimizing feature selection so that the classification results are more accurate and interpretative.

REFERENCES

Agung Prabowo, D., & Sudianto. (2023). Analisis Sentimen Sepak Bola Indonesia pada Twitter menggunakan K-Nearest Neighbors dan *Random Forest*. JSAI (Journal Scientific and Applied Informatics), 6(2), 217–227. https://doi.org/10.36085/jsai.v6i2.5337

Aufa, M. J., & Qoiriah, A. (2022). Analisis Sentimen Pengguna Platform Belajar Online Coursera menggunakan *Random Forest* dengan Metode Ekstraksi Fitur Word2vec. Journal of Informatics and Computer Science (JINACS), 244-255. https://doi.org/10.26740/jinacs.v4n02.p244-255

Baskoro, B. B., Susanto, I., & Khomsah, S. (2021). Analisis sentimen pelanggan hotel di purwokerto menggunakan metode *Random Forest* dan tf-idf (studi kasus: Ulasan pelanggan pada situs tripadvisor). *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, *3*(2), 21-29.J. K. Penulis, "Judul disertasi," disertasi doctor/Ph.D, Departemen, Universitas, Negara, tahun. https://doi.org/10.20895/INISTA.V3

Chamekh, A., Mahfoudh, M., & Forestier, G. (2022, July). Sentiment analysis based on deep learning in e-commerce. In International Conference on Knowledge Science, Engineering and Management (pp. 498-507). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-10986-7_40

Isnain, A. R., Supriyanto, J., & Kharisma, M. P. (2021). Implementation of K-Nearest Neighbor (K-NN) algorithm for public sentiment analysis of online learning. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *15*(2), 121-130.Penulis. (bulan, tahun). Judul. Dipresentasikan di Nama Konferensi. [Tipe media]. Tersedia: site/path/file https://doi.org/10.22146/ijccs.65176

Mehta, P., & Pandya, S. (2020). A review on sentiment analysis methodologies, practices and applications. International Journal of Scientific and Technology Research, 9(2), 601-609.

Nufus, R. H., & Surapati, U. (2024). Analisis Sentimen Persepsi Masyarakat Terhadap Timnas Indonesia U-23 dalam AFC-23 Asian Cup 2024 Pada Media Sosial X Menggunakan Metode Naïve Bayes Classifier. Jurnal Indonesia: Manajemen Informatika dan Komunikasi, 5(3), 2647-2657. https://doi.org/10.35870/jimik.v5i3.964

Rachman, F. F., & Pramana, S. (2020). Analisis sentimen pro dan kontra masyarakat Indonesia tentang vaksin COVID-19 pada media sosial Twitter. *Indonesian of Health Information Management Journal (INOHIM)*, *8*(2), 100-109. https://inohim.esaunggul.ac.id/index.php/INO/article/view/223/175

Sumantri, G., & Marwoto, B. S. H. (2024). (2024). Sumantri, G., & Marwoto, B. S. H. (2024). ANALISIS SENTIMEN DI TWITTER TERKAIT TIM NASIONAL SEPAK BOLA INDONESIA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE. Jurnal Kajian dan Terapan Matematika, 10(2), 96-104. 10, 96–104. 10.21831/jktm.v10i2.19561

Supriyadi, R., Gata, W., Maulidah, N., & Fauzi, A. (2020). Penerapan Algoritma *Random Forest* Untuk Menentukan Kualitas Anggur Merah. *E-Bisnis: Jurnal Ilmiah Ekonomi Dan Bisnis*, *13*(2), 67-75. https://doi.org/10.51903/e-bisnis.v13i2.247