JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 7, No. 1. December 2024 DOI: https://doi.org/10.34288/jri.v7i1.353
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

SENTIMENT ANALYSIS OF TWITTER DATA ON KIP-KULIAH USING TEXTBLOB AND GRADIENT BOOSTING

Nanang Ruhyana[-1], Desi Masdin Dama[-2*)]

Sains Data
Fakultas Teknologi Informasi
Universitas Nusa Mandiri
nanang.ngy@nusamandiri.ac.id[-1], 15210007@nusamandiri.ac.id[-2*)]

(*) Corresponding Author

Abstract

The Indonesian government aims to position the country among developed nations by 2045, with a primary focus on improving education quality from elementary to higher education levels. One of the key initiatives is the KIP-Kuliah (Indonesia Smart College Card) program, which supports high-achieving students from underprivileged economic backgrounds in accordance with UU No. 12/2012 on Higher Education. This study applies sentiment analysis using TextBlob and the Gradient Boosting algorithm to build a predictive model that identifies public support for the program through Twitter data. The results reveal a significant dominance of negative sentiment, with the model achieving an accuracy of 97%. These findings underscore the importance of sentiment analysis as a feedback tool for policymakers during the implementation of education-related programs. Furthermore, the results suggest that continuous monitoring of public opinion via social media can contribute to more adaptive and responsive policy development. This research highlights the need for future studies to expand the scope of analysis using more advanced natural language processing techniques for deeper understanding and broader coverage of public sentiment.

Keywords : *Higher education; KIP-Kuliah; Sentiment analysis; TextBlob; Gradient Boosting*

Abstrak

Pemerintah Indonesia memiliki visi untuk menjadikan Indonesia sebagai negara maju pada tahun 2045 dengan salah satu fokus utamanya yaitu peningkatan kualitas pendidikan dari jenjang dasar hingga perguruan tinggi. Salah satu program yang diluncurkan untuk mendukung tujuan tersebut adalah Kartu Indonesia Pintar Kuliah (KIP-Kuliah), yang ditujukan bagi mahasiswa berprestasi dari latar belakang ekonomi kurang mampu, sesuai dengan UU No. 12 Tahun 2012 tentang Pendidikan Tinggi. Penelitian ini bertujuan untuk memahami sentimen masyarakat terhadap program KIP-Kuliah melalui data Twitter, dengan menerapkan metode analisis sentimen berbasis TextBlob dan membangun model prediksi menggunakan algoritma Gradient Boosting. Hasil penelitian menunjukkan bahwa sentimen negatif mendominasi percakapan publik terkait program ini, dengan akurasi model mencapai 97%. Temuan ini menekankan pentingnya analisis sentimen dalam memberikan masukan kepada pemerintah dalam pelaksanaan kebijakan pendidikan. Selain itu, hasil ini menunjukkan perlunya pemantauan opini publik secara berkelanjutan melalui media sosial untuk mendukung perbaikan kebijakan yang lebih responsif. Penelitian selanjutnya disarankan untuk memperluas cakupan data dan menggunakan pendekatan NLP yang lebih canggih guna mendapatkan pemahaman yang lebih komprehensif terhadap respons masyarakat.

Kata kunci*: Pendidikan tinggi; KIP-Kuliah; Analisis sentimen; TextBlob; Gradient Boosting*

INTRODUCTION

One of the Indonesian government's key agendas to become a developed nation by 2045 is formulating a clear vision and the best strategies to improve the quality of human resource education from elementary levels to higher education. Beyond being a foundational step toward national prosperity in 2045, every Indonesian citizen has an equal right to access education, as stated in Article 31, paragraph 1 of the 1945 Constitution: "Every citizen has the right to education." Based on this mandate, the government launched the KIP-Kuliah (Indonesia Smart College Card) program to ensure that youth from economically disadvantaged backgrounds—especially those with strong academic achievements—can continue their studies in higher education without financial barriers (Izzhulhaq & Trisnaningsih, 2022).

KIP-Kuliah, aligned with UU No. 12 of 2012 on Higher Education, represents the Indonesian government's commitment to expanding access and opportunities for learning at the tertiary level while preparing a smart and competitive young generation. This scholarship aims to motivate students to maintain and improve their academic performance over time (Marita & Prayogi, 2024). To support this initiative, the government fully subsidizes tuition fees and provides living allowances to meet students' daily needs. Accordingly, to ensure the scholarship is used effectively and efficiently, recipients of KIP-Kuliah must comply with the established requirements and avoid wasteful spending so that the allocated funds are used wisely (Faadhilah et al., 2023).

Based on these considerations, this study focuses on the application of text mining for sentiment analysis to gain deeper insights for decision-making and public understanding of the KIP-Kuliah program, based on user comments from the X platform. This research utilizes TextBlob, a Python library for sentiment processing and analysis, alongside the Gradient Boosting algorithm to build a predictive model. Through TextBlob, the study extracts opinions and sentiments from textual data—specifically public reactions regarding the eligibility of KIP-Kuliah recipients. This study concentrates only on positive and negative sentiment labels to provide clearer and deeper insights, simplifying the analytical process and facilitating easier interpretation of results. This approach has also been shown to be effective in previous studies (Kurniawan & Susanto, 2019), where neutral sentiment often failed to contribute significant value to overall sentiment insights.

The Gradient Boosting algorithm is then employed to process the sentiment analysis results and develop a model capable of predicting public support or opposition toward the KIP-Kuliah program based on relevant features. This approach not only enables an evaluation of the program's feasibility from a public perspective but also allows stakeholders to gain a deeper understanding of how the program is received and responded to by the community—an essential component in shaping and implementing effective public policy (Aidah, 2022)

LITERATURE REVIEW

In the context of sentiment analysis regarding the eligibility of KIP-Kuliah scholarship recipients, this study employs TextBlob to perform natural language processing tasks such as tokenization and sentiment analysis on data collected from the X application. Additionally, the Gradient Boosting algorithm is applied to develop a predictive model capable of processing the sentiment analysis results generated by TextBlob, aiming to predict and evaluate public sentiment towards the KIP-Kuliah program.

A. Text Mining

Text mining is a process of extracting valuable information from unstructured text by identifying patterns, trends, and insights from text documents using various computational techniques and algorithms. This technique is commonly used to uncover hidden information, understand sentiment, perform topic analysis, and group texts based on characteristics such as themes or writing styles. In this study, text mining is applied to detect emotions in social media texts—specifically from the X application—to explore and identify meaningful patterns in textual data that are not available in structured database records (I Komang Dharmendra et al., 2023).

B. Sentiment Analysis

Sentiment analysis is a key aspect of technological development that uses a text mining approach to extract and determine sentiments and emotions contained within textual data. This process is widely applied to social media content, product reviews, and news articles to identify whether the opinions expressed are positive, negative, or neutral. It involves techniques such as natural language processing (NLP), text processing, and machine learning to classify text based on emotional expressions, evaluations, or the author's attitude (Wijaya & Panjaitan, 2024).

C. TextBlob

TextBlob is a Python library used for text processing in both Python 2 and 3. It provides an application programming interface (API) for various NLP tasks, including noun phrase extraction, sentiment analysis, classification, translation, and more. The output objects from

TextBlob are used in natural language learning applications. However, it should be noted that TextBlob currently only supports English-language recognition (Vonega et al., 2022).

D. **Gradient Boosting**

Gradient Boosting is well-known for its capability to handle complex datasets and deliver high accuracy in various tasks such as classification, regression, and ranking. To implement the Gradient Boosting model, the GradientBoostingClassifier module from the sklearn.ensemble library is used. The model is initialized and trained using training data (X_train and y_train). Subsequently, the model is employed to make predictions on test data (X_test), and its performance is evaluated using the accuracy_score metric (Hanif & Muntiari, 2024).

Previous Research

A study conducted by Novita Ranti Muntiari, Kharis Hudaiby Hanif, and Indah Chairun Nisa (Hanif & Muntiari, 2024) compared the performance of Logistic Regression, Support Vector Machine (SVM), and Gradient Boosting algorithms in sentiment analysis of student comments. The study utilized the Digital Teacher Assessment (DITA) application to evaluate teaching performance in schools. Student comments were classified into negative, positive, and neutral categories using lexicon-based sentiment analysis and TF-IDF. The results showed that Gradient Boosting achieved the highest accuracy of 97.5% and an F1-score of 97%, outperforming SVM (95%) and Logistic Regression (96%), indicating that Gradient Boosting is more effective in analyzing sentiment in student feedback.

Another study (Digno et al., 2023) by Christopher Digno, Muhammad Iqbal Jauhar, and Muhammad Nur Syaifullah from the Informatics Study Program, Faculty of Information Technology and Data Science, Universitas Sebelas Maret, Indonesia, explored the application of Deep Learning and Gradient Boosting in predicting Airbnb property prices based on sentiment analysis of user reviews. The data used in the study was sourced from Inside Airbnb for New York City and included property features and user reviews. By implementing a Gradient Boosting model, the study achieved the lowest mean squared error (MSE) of 0.1414. Moreover, the researchers compared Lasso and ElasticNet feature selection methods, with ElasticNet reducing the MSE from 0.1471 to 0.1370. This study contributes significantly to overcoming challenges in Airbnb rental price prediction using effective machine learning approaches that can be further explored.

RESEARCH METHODS

Types of research

This study uses a **quantitative approach** by utilizing a dataset extracted from Twitter related to the topic “Mahasiswa KIPK.” The focus of the study is on two primary categories of comments, namely positive and negative comments, which reflect public responses to the use of funds by KIP Kuliah scholarship recipients. By selecting 1,639 representative comments, the study aims to deeply analyze public sentiment regarding the program.

Time and place of research

The research was conducted in 2025 using publicly available Twitter data. Data processing and analysis were performed using Jupyter Notebook, an open-source web application for data analysis, modeling, machine learning, and education purposes.

Research target / subject

The research target is the collection of public comments related to KIP Kuliah recipients on the social media platform X (formerly known as Twitter). The dataset was selected based on relevance to the topic and its ability to represent a diverse range of opinions. The sampling was carried out purposively, focusing only on comments containing sentiments either positive or negative toward the KIP Kuliah program.

Procedure

The research procedure began with the extraction of relevant comments about KIP Kuliah scholarship recipients. The collected comments were then processed through several preprocessing stages, including:

1. **Case folding**: converting all text into lowercase.
2. **Tokenizing**: breaking down the text into tokens or individual words.
3. **Filtering**: cleaning the text by removing punctuation marks, stop words, and other irrelevant elements.

JURNAL RISET INFORMATIKA
Vol. 7, No. 1. December 2024
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

After preprocessing, the cleaned dataset was downloaded and prepared for the labeling stage. Three different methods were tried for labeling, with TextBlob being selected based on achieving the highest accuracy. The final labeled dataset was used to build a predictive model using the Gradient Boosting algorithm.

Data, Instruments, and Data Collection Techniques

The type of data used in this research is textual data in the form of comments on Twitter. The main instrument for data processing was Jupyter Notebook, which enabled text preprocessing, labeling, and modeling. Data collection was conducted through manual extraction and web scraping methods focused on comments related to KIP Kuliah scholarship recipients.

Data analysis technique

The analysis of the data involved several stages: text preprocessing, sentiment labeling using TextBlob, and model building using Gradient Boosting. The model's performance was evaluated based on accuracy metrics to determine the effectiveness of the sentiment analysis approach.

RESULTS AND DISCUSSION

After conducting a thorough review of the requirements necessary for the research process as outlined in the literature review, this section presents the results obtained from the data analysis and provides an in-depth discussion of the findings. The dataset was extracted from application X, resulting in a collection of 1,639 comments comprising 15 attributes, of which 7 are of integer (numerical) type and 8 are of string (textual) type.

Prepocessing Dataset

The initial stage of preprocessing began with installing the necessary libraries required for the preprocessing steps and reading the dataset, followed by displaying the top rows.

```
In [1]: #import Library

        import pandas as pd
        import re
        import string
        import nltk

In [3]: df = pd.read_csv("Mahasiswa_KIPK.csv")
        df.head(5)
Out[3]:
```

	conversation_id_str	created_at	favorite_count	full_text	id_str	image_url	in_reply_to_screen_name	lang	location	quote_count
0	1786410129799495830	Fri May 03 14:58:59 +0000 2024	0	Kemendikbud Imbau Pihak Kampus Mahasiswa dan M...	1786410129799495830	NaN	NaN	in	Bogor, Jawa Barat	0
1	1785649198622446038	Fri May 03 13:18:56 +0000 2024	0	@undipmenfess Lah masa gak boleh kuliah. Suruh...	1786384950859874425	NaN	undipmenfess	in	HLePS	0
2	1785935279276302350	Fri May 03 13:10:46 +0000 2024	0	@collegemenfess Sbg contoh di Solo kipk living...	1786382897676775539	NaN	collegemenfess	in	Indonesia	0
3	1785935279276302350	Fri May 03 13:09:10 +0000 2024	0	@ethper @Prasekti21 @collegemenfess Sbg contoh...	1786382493433893243	NaN	ethper	in	Indonesia	0
4	1786054457559756888	Fri May 03 13:08:33 +0000 2024	0	@unjsecret Kamu lihat update mahasiswa KIPK se...	1786382338903179747	NaN	bergulingan	in	NaN	0

Figure 1. Import Python Libraries and Read dataset

After reading the data and displaying the first five rows, the next step was to identify the data types of the 15 attributes available in the dataset.

After reading the data and displaying the first five rows, the next step was to identify the data types of the 15 attributes available in the dataset.

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1639 entries, 0 to 1638
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   conversation_id_str      1639 non-null   int64
 1   created_at               1639 non-null   object
 2   favorite_count           1639 non-null   int64
 3   full_text                1639 non-null   object
 4   id_str                   1639 non-null   int64
 5   image_url                104 non-null    object
 6   in_reply_to_screen_name  972 non-null    object
 7   lang                     1639 non-null   object
 8   location                 863 non-null    object
 9   quote_count              1639 non-null   int64
 10  reply_count              1639 non-null   int64
 11  retweet_count            1639 non-null   int64
 12  tweet_url                1639 non-null   object
 13  user_id_str              1639 non-null   int64
 14  username                 1639 non-null   object
dtypes: int64(7), object(8)
memory usage: 192.2+ KB
```

Figure 2. Description of Data Types and Attributes

Subsequently, to ensure that there were no duplicate entries, the "drop_duplicates" function was utilized.

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1639 entries, 0 to 1638
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   conversation_id_str      1639 non-null   int64
 1   created_at               1639 non-null   object
 2   favorite_count           1639 non-null   int64
 3   full_text                1639 non-null   object
 4   id_str                   1639 non-null   int64
 5   image_url                104 non-null    object
 6   in_reply_to_screen_name  972 non-null    object
 7   lang                     1639 non-null   object
 8   location                 863 non-null    object
 9   quote_count              1639 non-null   int64
 10  reply_count              1639 non-null   int64
 11  retweet_count            1639 non-null   int64
 12  tweet_url                1639 non-null   object
 13  user_id_str              1639 non-null   int64
 14  username                 1639 non-null   object
dtypes: int64(7), object(8)
memory usage: 192.2+ KB
```

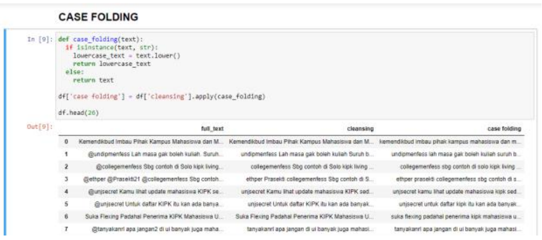
Figure 3. Removing Duplicate Data

The next stage involved cleaning the text within the DataFrame by removing URLs, HTML tags, emojis, symbols, and numbers. Each step was carried out sequentially using functions implemented with regular expressions (regex), resulting in cleaner text for further analysis.

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i1.353
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Figure 4. Removing URL, HTML Tags, Emojis, Symbols, and Numbers

After cleaning unnecessary symbols, numbers, and HTML tags, the case-folding function was applied to convert all letters in the text to lowercase. This step helps standardize the text format for consistency during analysis. The process was applied to the "cleansing" column, and the results were stored in the "case folding" column within the DataFrame.



CASE FOLDING

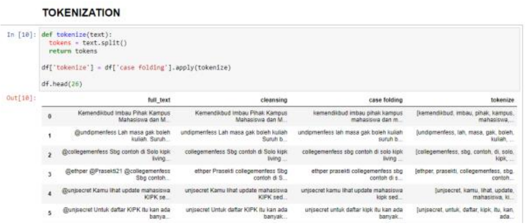
```
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case folding'] = df['cleansing'].apply(case_folding)

df.head(20)
```

Figure 5. Case Folding Implementation

The next step was the application of the tokenization function, which separates the text into a list of words (tokens) by splitting based on spaces. This process was applied to the "case folding" column, and the results were stored in the "tokenize" column within the DataFrame.



TOKENIZATION

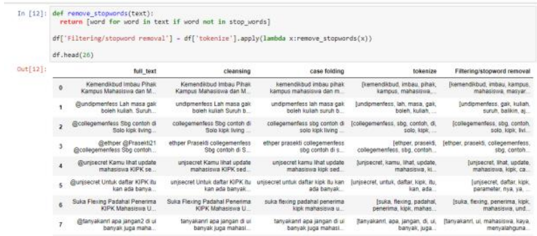
```
def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['case folding'].apply(tokenize)

df.head(20)
```

Figure 6. Tokenization Implementation

Subsequently, the remove_stopwords function was used to eliminate common words (stopwords) from the tokenized text. This was accomplished by filtering out words that appeared in the stopwords list. The results were stored in the "Filtering/stopword removal" column within the DataFrame.



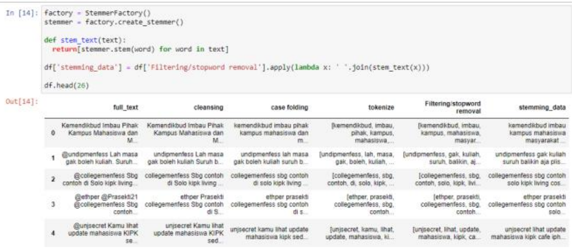
```
def remove_stopwords(text):
    return [word for word in text if word not in stop_words]

df['Filtering/stopword removal'] = df['tokenize'].apply(lambda x:remove_stopwords(x))

df.head(20)
```

Figure 7. Stopwords Removal Impelentation

In the final stage, stemming was applied to convert words into their root forms by utilizing the Sastrawi library. The stem_text function iterated through the words in the text that had already been filtered from stopwords, transformed each word into its root form, and stored the results in the "stemming_data" column within the DataFrame.



```
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    return[stemmer.stem(word) for word in text]

df['stemming_data'] = df['Filtering/stopword removal'].apply(lambda x: ' '.join(stem_text(x)))

df.head(20)
```

Figure 8. Stemming Impelentation

Dataset Labeling

After completing the installation of the required libraries, displaying the top five rows of data, and identifying the data types of each attribute, the next step was to separate the dataset based on its labels into two categories: negative and positive labels.

Figure 9. Labeling Dataset

Subsequently, using the Matplotlib library in Python, a bar chart was generated, as shown in *Figure 10*, illustrating the results of the labeling performed by TextBlob. This chart provides a clear visualization of the distribution between positive and negative sentiments, based on the implemented labeling outcomes.

P-ISSN: 2656-1743 | E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i1.353

Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

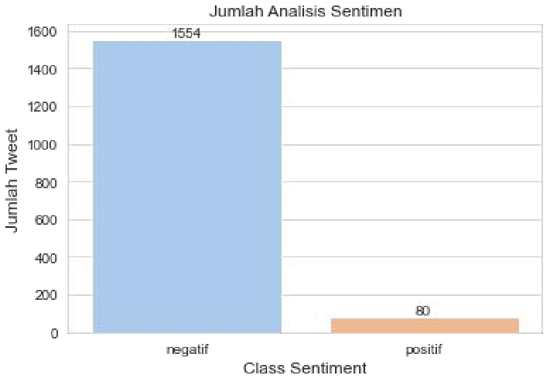


Figure 10. Result Visualization of labeling dataset

Based on the visualization results in *Figure 10*, it is evident that negative sentiment regarding the eligibility of KIP-Kuliah recipients is more dominant, with a total of 1,554 instances, while positive sentiment is recorded at only 80. This comparison indicates that the majority of responses or opinions collected tend to lean more toward negative sentiment than positive sentiment.

Dataset Modeling

In the final stage of this research, it is essential to ensure that the results from the TextBlob labeling are accurate and yield favorable outcomes. To achieve this, the author applied the Gradient Boosting algorithm as the primary choice due to its ability to generate the highest accuracy compared to other algorithms, with an accuracy score of 0.97. Furthermore, this algorithm also demonstrated excellent precision, recall, and F1-Score values, namely 0.97 for precision, 0.97 for recall, and 0.96 for F1-Score. Therefore, the use of the Gradient Boosting algorithm was deemed the most appropriate choice, as it provides consistent and reliable results in classifying sentiments in the text data that has been labeled with TextBlob.

```
# Membagi data menjadi data latih dan data uji dengan stratified split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Mengubah teks menjadi fitur numerik
vectorizer = CountVectorizer()
X_train_counts = vectorizer.fit_transform(X_train)
X_test_counts = vectorizer.transform(X_test)

# Gradient Boosting
gb = GradientBoostingClassifier()
gb.fit(X_train_counts, y_train)
y_pred_gb = gb.predict(X_test_counts)

# Evaluasi
accuracy = accuracy_score(y_test, y_pred_gb)
precision = precision_score(y_test, y_pred_gb, average='weighted')
recall = recall_score(y_test, y_pred_gb, average='weighted')
f1 = f1_score(y_test, y_pred_gb, average='weighted')

print("Gradient Boosting Accuracy: {:.2f}".format(accuracy))
print("Gradient Boosting Precision: {:.2f}".format(precision))
print("Gradient Boosting Recall: {:.2f}".format(recall))
print("Gradient Boosting F1-Score: {:.2f}".format(f1))

Gradient Boosting Accuracy: 0.97
Gradient Boosting Precision: 0.97
Gradient Boosting Recall: 0.97
Gradient Boosting F1-Score: 0.96
```

Figure 11. Result of modeling dataset

CONCLUSIONS AND SUGGESTIONS

Conclusion

The results of this research demonstrate that the use of TextBlob for sentiment analysis labeling and the Gradient Boosting algorithm for predictive modeling successfully revealed the dominance of negative sentiment from Twitter users regarding the KIP-Kuliah program. The model achieved high accuracy, with 97% accuracy, 97% precision, 97% recall, and 96% F1-Score. Through a comprehensive preprocessing process, including data cleaning, tokenization, and stemming, the data was validated as ready for in-depth analysis. The results from the TextBlob labeling, which identified 1,554 negative sentiments and 80 positive sentiments, emphasize the importance of sentiment analysis technology in providing a more comprehensive understanding of public opinion on government policies in education. This enables more effective evaluations by relevant stakeholders, as they receive more valid and specific information(Agustina et al., 2021; Arfyanti et al., 2022; Azzahrawani et al., 2024; DjajaPutra et al., 2020; Fauziyyah, 2020; Gagan Suganda et al., 2022; Hosseini et al., 2023; Khotimah et al., 2022; Kurniawijaya & Karsana, 2023; Marita & Prayogi, 2024; Nitha Kumala Dewi, 2023; Putri et al., 2021).

Suggestion

Based on the sentiment analysis results of this research, it is recommended that the government and program administrators of KIP-Kuliah consider the findings to iterate and improve the program's implementation. Additionally, they should continue to monitor public sentiment via social media platforms to gather timely and relevant feedback, allowing for early identification and resolution of issues. This will help gain more positive support for the program from the public. Furthermore, future research is encouraged to expand the scope of analysis by incorporating and analyzing more data sources, as well as utilizing more advanced NLP (Natural Language Processing) technologies to provide more detailed insights.

REFERENCES

Agustina, D. A., Subanti, S., & Zukhronah, E. (2021). Implementasi Text Mining Pada Analisis Sentimen Pengguna Twitter Terhadap Marketplace di Indonesia Menggunakan Algoritma Support Vector Machine. *Indonesian Journal of Applied Statistics*, *3*(2), 109. https://doi.org/10.13057/ijas.v3i2.44337

Aidah, N. A. (2022). Analisis Kebijakan Program Beasiswa Kartu Indonesia Pintar-Kuliah (Kip-K) Di Universitas Diponegoro. *Jurnal Ilmu Administrasi Dan Studi Kebijakan (JIASK)*, *5*(1), 1–22.

https://doi.org/10.48093/jiask.v5i1.91

Arfyanti, I., Fahmi, M., & Adytia, P. (2022). Penerapan Algoritma Decision Tree Untuk Penentuan Pola Penerima Beasiswa KIP Kuliah. *Building of Informatics, Technology and Science (BITS)*, *4*(3), 1196–1201. https://doi.org/10.47065/bits.v4i3.2275

Azzahrawani, N. R., Arkanudin, Alamri, A. R., Adha, N., Nuari, O. L., & Heronimus, V. (2024). Implementation of the Independent College KIP Policy at Tanjungpura University. *JKMP (Jurnal Kebijakan Dan Manajemen Publik)*, *12*(1), 58–68. https://doi.org/10.21070/jkmp.v12i1.1765

Digno, C., Jauhar, M. I., & Syaifullah, M. N. (2023). Pendekatan Deep Learning dan Gradient Boosting dalam Prediksi Harga Properti Airbnb dengan Analisis Sentimen. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, *12*(1), 191. https://doi.org/10.24843/jlk.2023.v12.i01.p22

DjajaPutra, I. O., Prilianti, K. R., & Tirma Irawan, P. L. (2020). Implementasi Text Mining Untuk Analisis Opini Masyarakat Terhadap Kinerja Layanan Transportasi Online Dengan Analisis Faktor. *Jurnal Simantec*, *8*(2), 45–53. https://doi.org/10.21107/simantec.v8i2.6764

Faadhilah, G., Gumilar, R., & Nurdianti, R. R. S. (2023). Pengaruh Lifestyle, Self Control, dan Financial Literacy terhadap Perilaku Konsumsi. *Global Education Journal*, *1*(3), 177–190. https://doi.org/10.59525/gej.v1i3.175

Fauziyyah, A. (2020). ANALISIS SENTIMEN PANDEMI COVID19 PADA STREAMING TWITTER DENGAN TEXT MINING PYTHON. *Jurnal Ilmiah SINUS*, *18*, 31. https://doi.org/10.30646/sinus.v18i2.491

Gagan Suganda, Marsani Asfi, Ridho Taufiq Subagio, & Ricky Perdana Kusuma. (2022). Penentuan Penerima Bantuan Beasiswa Kartu Indonesia Pintar (Kip) Kuliah Menggunakan Naïve Bayes Classifier. *JSiI (Jurnal Sistem Informasi)*, *9*(2), 193–199. https://doi.org/10.30656/jsii.v9i2.4376

Hanif, K. H., & Muntiari, N. R. (2024). Penerapan Algoritma Decision Tree, Svm, Naive Bayes Dalam Deteksi Stunting Pada Balita. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, *8*(1), 105–109. https://doi.org/10.46880/jmika.vol8no1.pp105-109

Hosseini, P., Khoshsirat, S., Jalayer, M., Das, S., & Zhou, H. (2023). Application of text mining techniques to identify actual wrong-way driving (WWD) crashes in police reports. *International Journal of Transportation Science and Technology*, *12*(4), 1038–1051. https://doi.org/10.1016/j.ijtst.2022.12.002

I Komang Dharmendra, Ricky Aurelius Nurtanto Diaz, Muhamad Samsudin, I Gusti Agung Ngurah Rai Semadi, & I Made Agus Wirahadi Putra. (2023). Text Mining Untuk Mendeteksi Emosi Pengguna Terhadap “Nusantara” Sebagai Nama Ikn. *Jurnal Teknologi Informasi Dan Komputer*, *9*(5), 457–463. https://doi.org/10.36002/jutik.v9i5.2639

Izzhulhaq, R. R., & Trisnaningsih, S. (2022). Analisis Implementasi Kebijakan Pro Poor Budgeting pada Program Kartu Indonesia Pintar Terhadap Penerima Program Kartu Indonesia Pintar (Studi Kasus Mahasiswa S-1 Akuntansi Universitas Pembangunan Nasional “Veteran” Jawa Timur Angkatan 2021). *J-MAS (Jurnal Manajemen Dan Sains)*, *7*(2), 523. https://doi.org/10.33087/jmas.v7i2.444

Khotimah, K., Anggraini, L. W., Alfirnanda, W. T., & Tahyudin, I. (2022). Decision Support System for Selecting KIP-K Recipients at Amikom University, Purwokerto Using the TOPSIS Method. *Internet of Things and Artificial Intelligence Journal*, *2*(4), 279–290. https://doi.org/10.31763/iota.v2i4.566

Kurniawan, I., & Susanto, A. (2019). Implementasi Metode K-Means dan Naïve Bayes Classifier untuk Analisis Sentimen Pemilihan Presiden (Pilpres) 2019. *Eksplora Informatika*, *9*(1), 1–10. https://doi.org/10.30864/eksplora.v9i1.237

Kurniawijaya, P. A., & Karsana, I. W. W. (2023). Implementasi Metode AHP Dalam Sistem Penunjang Keputusan Penerima KIP Kuliah. *JUKI : Jurnal Komputer Dan Informatika*, *5*(1), 22–31.

Marita, T., & Prayogi, A. (2024). Telaah Deskriptif Motivasi Berprestasi Mahasiswa Penerima Beasiswa Kartu Indonesia Pintar Kuliah (KIP-K). *RUKASI: Jurnal Ilmiah Perkembangan Pendidikan Dan Pembelajaran*, *1*(02), 54–64. https://doi.org/10.70294/rr80jk09

Nitha Kumala Dewi. (2023). Identifikasi Berita Hoax dengan Menerapkan Algoritma Text Mining. *Journal of Informatics, Electrical and Electronics Engineering*, *2*(3), 65–74. https://doi.org/10.47065/jieee.v2i3.888

Putri, N. A. Y., Subagio, R. T., & Asfi, M. (2021). Sistem Pendukung Keputusan Penilaian Kinerja Mahasiswa KIP Kuliah dengan Penerapan Metode TOPSIS dan PROMETHEE.

Jurnal Media Informatika Budidarma, *5*(4), 1394. https://doi.org/10.30865/mib.v5i4.3268

Vonega, D. A., Fadila, A., & Kurniawan, D. E. (2022). Analisis Sentimen Twitter Terhadap Opini Publik Atas Isu Pencalonan Puan Maharani dalam PILPRES 2024. *Journal of Applied Informatics and Computing*, *6*(2), 129–135. https://doi.org/10.30871/jaic.v6i2.4300

Wijaya, N., & Panjaitan, E. S. (2024). *Analisis Sentimen Ulasan Aplikasi Instagram di Google Play Store : Pendekatan Multinomial Naive Bayes dan Berbasis Leksikon*. *6*(2), 921–929. https://doi.org/10.47065/bits.v6i2.5615