JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 2, Juni 2025, Pp. 1286-1298

COMBINATION OF LOGISTIC REGRESSION AND NAÏVE BAYES IN SENTIMENT ANALYSIS OF ONLINE LENDING APPLICATION PLATFORMS BY UTILIZING THE LEXICONS FEATURE

Muhammad Faisal Zaenudin*[1)], Yuliant Sibaroni[2)]
1. Telkom University, Indonesia
2. Telkom University, Indonesia

Article Info
Keywords: Sentiment Analysis; Logistic Regression; Naïve Bayes; Lexicon; Online Loans

Article history:
Received 11 November 2024
Revised 15 Desember 2024
Accepted 14 Januari 2025
Available online 15 Maret 2025

DOI :
https://doi.org/10.29100/jipi.v10i2.6163

* Corresponding author.
Corresponding Author
E-mail address:
isalza@student.telkomuniversity.ac.id

ABSTRACT
In the digital age, online lending apps have become an important tool in facilitating financial transactions and supporting MSMEs. However, the existence of negative opinions related to violations such as theft of customer data raises concerns in the community. This research aims to analyze sentiment towards online loan applications, especially Kredivo, using a combination of Logistic Regression and Naïve Bayes which is optimized through the Lexicons feature. Data is taken from Google Play Store reviews, then labeling, preprocessing, and feature extraction are executed through TF-IDF technique. The classification models built are Naive Bayes (NB) and Logistic Regression (LR), where the results of the two models are combined with the ensemble voting method using lexicons features. The evaluation results show that the combination approach of the three methods can significantly improve classification accuracy compared to the use of a single method. The combined model achieved an accuracy of 89.62%, higher than Logistic Regression (86.19%) and Naive Bayes (83.54%).

.

I. INTRODUCTION

IN the era of digitalization, the internet is something that cannot be separated from human life. The emergence of various technological developments, one of which is online lending. Online lending is the concept of online money lending to facilitate transactions [1]. Online loans also help finance MSMEs to develop businesses. However, the convenience provided causes violations such as theft of customer data. Opinions or opinions expressed regarding these online loans vary. Public opinion related to online loans varies, not yet known including negative or positive opinions. Therefore, it is necessary to conduct sentiment analysis to help people know the sentiment class in tweet data so that people are wiser in using online loan platforms. One way to process these reviews is to use sentiment analysis, which classifies opinions into sentiments that are positive or negative [2][3].

A study on sentiment analysis of PPKM policies during the COVID-19 pandemic using 3,516 tweets collected during that time. For this analysis, the K-Nearest Neighbor (KNN) and Naïve Bayes Classifier (NBC) algorithms were used; both showed an accuracy of 79.67% and 78.86%, respectively[4]. Sentiment analysis research on MyPertamina application reviews on Google Playstore using the Naïve Bayes Classifier algorithm shows 87% accuracy, 86% precision, 90% recall, and f1 score[5]. Sentiment analysis research on online loans (pinjol) was also conducted using the Naïve Bayes Classifier algorithm on 650 data covering positive, negative, and neutral sentiments. The data is divided into two parts: 80% for training and 20% for testing, resulting in an accuracy of 75%[6].

Sentiment analysis research on Twitter using Logistic Regression on 349 tweets (177 positive, 172 negative) with a division of 80:20 training and testing data resulted in 78.57% accuracy, 76.92% precision, 83.3% recall, and 80% F1-score [7]. Research on sentiment analysis of digital population application reviews, the Logistic Regression algorithm obtained an accuracy of 78.83%, recall 55.84%, precision 71.63%, and F1-score 55.34%. In the meantime, The accuracy of the K-Nearest Neighbor method was 82.43%, recall 65.97%, precision 77.88%, and F1-score 68.72%. These results show that the K-Nearest Neighbor algorithm has better accuracy performance with 82.43%, 3.60% higher than Logistic Regression, so the k-nearest neighbor algorithm is better than the logistic regression algorithm[8].

Another study on sentiment analysis with the level of user satisfaction of Indonesian Cellular Telecommunications Service Providers on Twitter using the Support Vector Machine method and lexicon-based features resulted in an accuracy value of 79%, precision of 65%, recall 97%, and f-measure 78%. Research by Lestari et al. (2019) using the Learning Vector Quantization (LVQ) method and lexicon-based features for clickbait video classification on YouTube shows that the use of lexicon-based features can significantly improve system accuracy, from 54.54% to 90.91%, precision from 1 to 85.71%, recall remains 1, and f-measure from 28.58% to 92.31%. In conclusion, the addition of lexicon-based features to the Naïve Bayes and Logistic Regression algorithms has the potential to improve sentiment analysis performance[9].

Based on the results of previous research, various sentiment analysis techniques have been used, and the results vary. K-Nearest Neighbor (KNN), Naïve Bayes Classifier (NBC), and Logistic Regression are the three main methods often used. A simple algorithm known as KNN classifies data based on the majority of the classes of its k nearest neighbors. This method is effective for small to medium datasets and requires no assumptions on the data distribution; however, on large datasets that are sensitive to unimportant elements, its performance may degrade. In contrast, NBC is a fast and efficient probabilistic classification method that can handle small datasets with many features. However, the assumption that the features are independent is often unrealistic, and if this assumption is violated, the performance may degrade. Logistic Regression can model the likelihood of class membership and handle both categorical and continuous features and is relatively resistant to overfitting. However, this method requires more data to obtain stable results and is less effective for complex non-linear problems.

For this research, the method chosen was a combination of Logistic Regression and Naïve Bayes, as well as the application of group voting methods with lexicon features. There are a number of factors that influenced the choice of this approach. The combination firstly capitalizes on the strengths of both algorithms: Logistic Regression offers high probabilities, while Naive Bayes is very effective in handling a wide range of features. Secondly, previous research has shown that the use of lexicon features can significantly improve accuracy. For several reasons, this combination method is expected to improve the accuracy of sentiment analysis compared to previous studies. Combining the Power of Algorithms: Logistic Regression is well-known for its ability to handle both categorical and continuous features. In contrast, Naive Bayes is a fast and effective probabilistic classification method that handles many features and is suitable for small to medium datasets. One of the disadvantages of one algorithm can be compensated by the advantages of the other algorithm when these two algorithms are combined. Use of Lexicon Features: In previous research, lexicon features have been shown to significantly improve model performance; they incorporate information about sentiment from specific vocabulary that is frequently used in reviews, which can improve the accuracy of sentiment classification. Ensemble Voting Method: The model can combine predictions from both algorithms to make a more accurate final decision by using the ensemble voting method. This method can improve the overall accuracy and reduce bias and variance by taking the majority vote of the Logistic Regression and Naive Bayes predictions.

The selection of the Kredivo online loan application as a case study in this research is based on several key factors. Kredivo is one of the leading online lending platforms in Indonesia with a large and diverse user base, thus providing a rich dataset for sentiment analysis. As a fast-growing fintech service, Kredivo generates a significant volume of reviews on the Google Play Store, thus enabling sufficient data collection for comprehensive analysis. The diversity of Kredivo's loan products, ranging from short-term loans to installments, triggers diverse user responses, creating interesting complexities for sentiment analysis. In addition, as a major player in the fintech industry, sentiment analysis of Kredivo can provide valuable insights into people's perceptions of online loan services in general, these characteristics make Kredivo an ideal subject for this research. The results obtained from this research are expected to help the field of sentiment analysis, especially in determining the level of user satisfaction with Kredivo online loan services.

II. METHODS

Figure 1 shows the stages of the method in building sentiment analysis. The initial stage of review data collection (scrapping) is taken from reviews on the Kredivo application in the Google Play Store which will be used as a dataset, then the second stage is the process of manually labeling the dataset with 2 types of labels, namely positive and negative. The dataset is then divided into three stages: 20% is used for test data, and the remaining 80% is used for training data. These proportions are used depending on several factors. For example, a ratio of 70:30 or 90:10 is also often used; a 70:30 ratio may provide more data for training on very large datasets, but may reduce the accuracy of the model's assessment on test data.

The model is provided with enough diverse data to identify the underlying pattern thanks to the 80% allocation of data for training. The model will function better if more data is provided for training. Secondly, by allocating 20% of the information, we can assess the model's own performance as a test. This strategy is supported by a study conducted by [10] and one study explaining the efficacy of the 80:20 split ratio by [2]. The dataset then goes through preparation in the third stage, when the dataset is filtered or cleaned to produce clean and high-quality data. Each clean dataset then undergoes feature extraction or word weighting using the TF-IDF algorithm in the fourth stage of preparation. In the fifth stage, Multinomial Naive Bayes and Logistic Regression methods are used in the classification stage based on the feature extraction results. Furthermore, in the sixth stage, labeling is done using a dataset that has been cleaned in the preprocessing process with the lexicon method using vander sentiment where all data is translated into English. The final step is to acquire the voting results from the three in the form of prediction labels from each test set by combining the outcomes of Multinomial Naive Bayes, Logistic Regression, and Lexicons with the voting method.

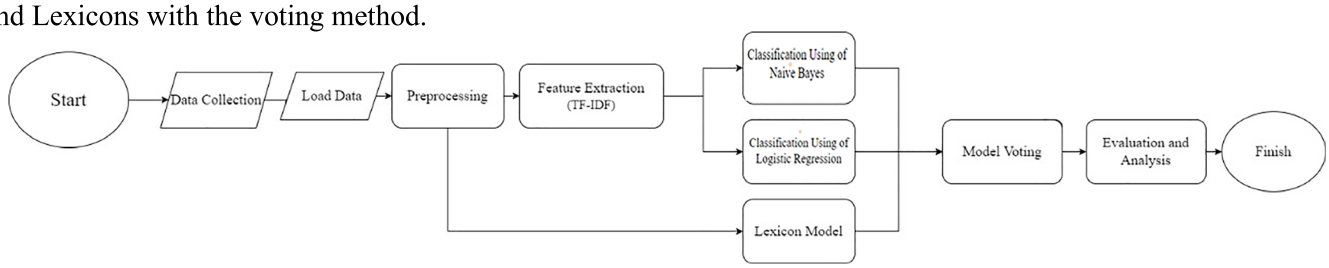

Fig. 1. Process Flow

A. Data Collection

The dataset used in this study was taken from reviews of the Kredivo online loan application on the Goole Playstore which consists of 10,000 rows of data with four attributes. This dataset is then labeled with Manual Annotation or manually by reading the reviews one by one and then determining whether the reviews contain positive or negative sentences, as shown in Table 1.

TABLE I
DATASET

UserName	Score	Content	Label
Justin Lionel	5	good	Positive
cts invory	5	Setelah di upgrade malah g bisa di aplikasi,,, aneh.. Skr udah bisa di buka	Negative
Alma Amira	5	Pokok nya the best banget kredivo mantap lah limit nya sdh naik bulan ini, Terimakasih kredivo semoga makin sukses	Positive
Dheni Ajjah	5	Kredivo sudah saya kasih bintang 5 Tapi tolong ya limit nya di naekin lagi jgn segituÂ² aja Saya bayar lancar ga pernh telat Saudah upgarde premium tapi limit nya segituÂ² aja ga naekÂ²	Positive

B. Preprocessing

The review data is cleansed to exclude unnecessary words and emoticons during the preprocessing phase. This step will yield review material that is more logically organized, or what is known as "clean text." The following is the order in which the preprocessing stage is completed:

a. Case Folding Aiming to make all letters lowercase, case folding makes words that contain capital and non-capital letters equal.
b. Stopword uses sastrawi to eliminate common terms that appear frequently but have little significance.

TABLE II
STOPWORD DICTIONARY

No	Stopword	No	Stopword	No	Stopword	No	Stopword	No	Stopword	No	Stopword
1	yang	26	sehingga	51	kami	76	dalam	101	namun	113	di
2	jika	27	belum	52	itu	77	saya	102	ini	114	atau
3	setelah	28	adalah	53	sudah	78	itulah	103	serta	115	agar
4	ketika	29	mereka	54	daripada	79	guna	104	bahwa	116	tentang
5	ada	30	yakni	55	supaya	80	seraya	105	secara	117	sampai
6	yaitu	31	sesudah	56	seolah	81	demikian	106	maka	118	agak
7	sebelum	32	selain	57	anu	82	setidaknya	107	pun	119	nggak
8	sebab	33	dulunya	58	setiap	83	pun	108	tanpa	120	saja
9	dahulu	34	sebetulnya	59	bagaimanapun	84	para	109	ingin	121	antara
10	seharusnya	35	apalagi	60	ke	85	tidak	110	pasti	122	kepada

11	amat	36	pada	61	dan	86	bagi	111	menurut	123	dari
12	unttuk	37	kembali	62	sekitar	87	bisa	112	karena	124	hanya

c. Tokenizing aims to break down sentences into individual words or tokens.
d. Stemming aims to change the affixed word to its basic word form using an Indonesian stemmer, namely sastrawi.

TABLE III
STEMMING

Input	Output	Description
[‘cepat’,’praktis’,’menyenangkan’]	cepat praktis senang	in this stemming stage, the affixed word is converted into its basic word form, such as the word menyenangkan into the word senang.
['trima', 'kasih', 'kredivo', 'bintang', 'aplikasi', 'terbaik', 'pilihan', 'pinjaman', 'dn', 'pembelian', 'yg', 'dgn', 'jangka', 'yg', 'pulsa', 'biasapulsa', 'datapulsa', 'listrikpinjaman', 'tunai', 'sd', 'pembelian', 'online', 'silakan', 'download', 'aplikasinya', 'dn', 'tdk', 'ribet', 'coba', 'biar', 'pahamthanks', 'kredivo']	trima kasih kredivo bintang aplikasi baik pilih pinjam dn beli yg dgn jangka yg pulsa biasapulsa datapulsa listrikpinjaman tunai sd beli online sila download aplikasi dn tdk ribet coba biar pahamthanks kredivo	In this stemming stage, the affixed word is converted into its base word form, such as the word pilihan into the word pilih.
['trima', 'kasih', 'kredivo', 'bintang', 'aplikasi', 'terbaik', 'pilihan', 'pinjaman', 'dn', 'pembelian', 'yg', 'dgn', 'jangka', 'yg', 'pulsa', 'biasapulsa', 'datapulsa', 'listrikpinjaman', 'tunai', 'sd', 'pembelian', 'online', 'silakan', 'download', 'aplikasinya', 'dn', 'tdk', 'ribet', 'coba', 'biar', 'pahamthanks', 'kredivo']	trima kasih kredivo bintang aplikasi baik pilih pinjam dn beli yg dgn jangka yg pulsa biasapulsa datapulsa listrikpinjaman tunai sd beli online sila download aplikasi dn tdk ribet coba biar pahamthanks kredivo	In this stemming stage, the affixed word is converted into its base word form, such as the word pinjaman into the word pinjam.
['trima', 'kasih', 'kredivo', 'bintang', 'aplikasi', 'terbaik', 'pilihan', 'pinjaman', 'dn', 'pembelian', 'yg', 'dgn', 'jangka', 'yg', 'pulsa', 'biasapulsa', 'datapulsa', 'listrikpinjaman', 'tunai', 'sd', 'pembelian', 'online', 'silakan', 'download', 'aplikasinya', 'dn', 'tdk', 'ribet', 'coba', 'biar', 'pahamthanks', 'kredivo']	trima kasih kredivo bintang aplikasi baik pilih pinjam dn beli yg dgn jangka yg pulsa biasapulsa datapulsa listrikpinjaman tunai sd beli online sila download aplikasi dn tdk ribet coba biar pahamthanks kredivo	In this stemming stage, the affixed word is converted into its base word form, such as the word pembelian into the word beli.
['bos', 'q', 'tolong', 'bantu', 'pembayaran', 'tagihan', 'sy', 'yg', 'terverifikasisy', 'bayar', 'tagihan', 'dr', 'tgl', 'kemarin', 'bukti', 'jg', 'sy', 'cantumkantraksaksi', 'pembayaran', 'jg', 'sukses']	bos q tolong bantu bayar tagih sy yg terverifikasisy bayar tagih dr tgl kemarin bukti jg sy cantumkantraksaksi bayar jg sukses	In this stemming stage, the affixed word is converted into its base word form, such as the word pembayaran into the word bayar.
['bos', 'q', 'tolong', 'bantu', 'pembayaran', 'tagihan', 'sy', 'yg', 'terverifikasisy', 'bayar', 'tagihan', 'dr', 'tgl', 'kemarin', 'bukti', 'jg', 'sy', 'cantumkantraksaksi', 'pembayaran', 'jg', 'sukses']	bos q tolong bantu bayar tagih sy yg terverifikasisy bayar tagih dr tgl kemarin bukti jg sy cantumkantraksaksi bayar jg sukses	In this stemming stage, the affixed word is converted into its base word form, such as the word tagihan into the word tagih.
['penggunaan', 'bukalapak', 'paylater', 'lancar', 'cepat', 'pinjaman', 'tunai', 'langsung', 'acc', 'terima', 'kasih', 'kredivo']	guna bukalapak paylater lancar cepat pinjam tunai langsung acc terima kasih kredivo	In this stemming stage, the affixed word is converted into its base word form, such as the word penggunaan into the word guna.
['kreditvo', 'skrng', 'sdah', 'update', 'limit', 'sy', 'rendah', 'skali', 'sy', 'bayar', 'seminggu', 'jatuh', 'tempo', 'tpi', 'limitnya', 'dikurangi', 'aneh']	kreditvo skrng sdah update limit sy rendah skali sy bayar minggu jatuh tempo tpi limit rang aneh	In this stemming stage, the affixed word is converted into its base word form, such as the word limitnya into the word limit.
['puas', 'membantu', 'untung', 'kekurangan', 'apapun']	puas bantu untung kurang apa	In this stemming stage, the affixed word is converted into its base word form, such as the word kekurangan into the word kurang.
['puas', 'membantu', 'untung', 'kekurangan', 'apapun']	puas bantu untung kurang apa	In this stemming stage, the affixed word is converted into its base word form, such as the word membantu into the word bantu.

C. Feature Extraction Term Frequency – Inverse Document Frequency (TF-IDF)

According to the TF-IDF technique, a word's weight indicates how relevant it is to a given document; the greater the weight, the more important a function the word plays in constructing the document. The weight of each word in a document, or even a collection of papers, is determined using the TF-IDF approach [3]. The following are the steps of TF-IDF [11]:

a. Count the instances of term i in the document j (tfi,j).
b. Determine how many documents use the phrase i (dfi).
c. Use the calculation to determine the inverse document frequency (IDF) weight value:

$$\mathrm{id}f_i = \log\left\{\left(\frac{N}{df_i}\right\}\right) \tag{1}$$

Description:
N = jumlah total dokumen

d. Calculate the TF-IDF weight value using the compound:

$$w_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{id}f_i \tag{2}$$

Deskripsi:
$w_{i,j}$ = weight of term i against document j
$\mathrm{t}f_{i,j}$ = frequency of term i in document j
$\mathrm{id}f_i$ = weight value of term IDF i

The data will next be divided into 80% train and 20% test portions after the labeling and preprocessing steps are completed. Words are then weighted using the TF-IDF approach. As stated by [12], The goal of this procedure is to extract a representation of each document's value from the training data. From there, a vector between the document and the term will be created, and the prototype vector will be used to calculate how similar the document is to the cluster. additionally known as the cluster centroid. After splitting the data, the train portion yielded 7535 with a weighting result of 11109, while the test portion yielded 1884 with the same weighting result.

TF-IDF was chosen for feature extraction because of its ability to give higher weights to terms that appear infrequently but are important in a particular document, while terms that appear frequently in many documents are given lower weights. This technique is very effective in dealing with texts that have many common words and diverse topics, such as in document analysis and text classification; it helps in finding words that are more relevant in the context of the document.

TF-IDF is different from other feature extraction methods, such as Bag of Words (BoW) or simple frequency-based methods. It takes into account how common or rare a word is in the entire document collection, not just its frequency in a particular document. This allows the model to distinguish between words that are highly specific and relevant to the document and words that are less informative because they appear frequently in many documents. As a result, when compared to conventional feature extraction methods, TF-IDF tends to provide a more significant feature representation and improve modeling performance and classification results.

D. Classification using of Naïve Bayes

Naïve Bayes is a simple probabilistic classification method, which calculates a set of probabilities by combining frequencies and combinations of values from a given dataset. The algorithm applies Bayes' theorem and makes the assumption that all attributes are independent or not interdependent, considering the values of the class variables. In the Naïve Bayes method, the probabilities involved are obtained through calculating the frequency of occurrence of certain attributes in the training data. Although the assumption that all attributes are independent in Naïve Bayes is often not met in real-world situations, this algorithm often yields good results in practice [13].

One of the Naive Bayes models for text classification is multinomial Naive Bayes. Multinomial Naïve Bayes is a supervised learning method, so each data needs to be labeled before training. The probability of a document d being in class c can be calculated using Equation (3) [14].

$$\mathrm{P}(\mathrm{c}|d) \propto \mathrm{P}(\mathrm{c}) \prod_{k=1}^{n} P(t_k|c) \tag{3}$$

Description:
$P(c|d)$:Probability of document d being in class c

$P(c)$:Prior probability of a document being in class c
$\{t1,t1,t1, \dots ,tn \}$:Tokens in document d that are part of the vocabulary with number n
$P(tk|c)$:Conditional probability of term tk being in document of class c

Finding the appropriate class for a document is the goal of document categorization. The best class in Naïve Bayes classification is determined by finding the maximum a posteriori (MAP) class cmap through Equation (4).

$$c_{map} = \underset{c\,\in C}{\arg\max}\, \hat{P}(c) \prod_{k=1}^{n} \hat{P}(t_k|c) \tag{4}$$

P is written with $\hat{P}$ because the actual values of $P(c|d)$ and $\hat{P}(t_k|c)$ are unknown, which will be calculated during the training process [14].

In equation (4), there are many conditional probabilities multiplied in the Multinomial Naïve Bayes process, which may cause floating point underflow problem. To solve this, it is better to do the summation on the logarithm of the probabilities. The class with the highest logarithm of probability will be the class with the best probability for that document. This follows the logarithm principle that log(xy) equals log(x) plus log(y). Equation (4) which uses the logarithm of probability can be expressed in the form of Equation (5) [14].

$$c_{map} = \underset{c\,\in C}{\arg\max}[\log \hat{P}(c) + \sum 1 \leq k \leq n \log \hat{P}(t_k|c)] \tag{5}$$

The probabilities $\hat{P}(c)$ and $\hat{P}(t_k|c)$ are obtained using the maximum likelihood method, which refers to the relative frequency of the parameters. To calculate the prior probability, Equation (6) can be used.

$$\hat{P}(c) \approx \frac{N_c}{N} \tag{6}$$

The probability $\hat{P}(t_k|c)$ is the relative frequency of term t in documents belonging to class c, and can be calculated using Equation (7).

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum t'\epsilon v^T ct'} \tag{7}$$

Description:

$\hat{P}(t|c)$:Conditional probability of term t being in document of class c
T_{ct} :Number of occurrences of term t in documents with category c
$\sum t'\epsilon v^T ct'$:Total frequency of all terms in class c

The maximum likelihood calculation has a weakness, where words that do not appear in the training data will have a probability value of zero. This results in the value of P(c|d) being zero, as multiplication by zero will result in zero. To overcome this problem, an add-one or Laplace smoothing technique is used, which turns Equation (8) into Equation (9).

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum t'\epsilon v(T_{ct} + 1)} = \frac{T_{ct} + 1}{(\sum t'\epsilon v T_{ct'}) + B'} \tag{8}$$

B : Number of all terms in the vocabulary

For the formulation of Multinomial Naïve Bayes using TF-IDF weighting, it can be seen in Equation (9) [15].

$$\hat{P}(t|c) = \frac{W_{ct} + 1}{(\sum t'\epsilon v T_{ct'}) + B'} \tag{9}$$

Description:

Wct: TF-IDF weight of term t in document with category c
$\sum w' \in V\ Wct'$: Sum of TF-IDF weights of all terms in class c
B': Total IDF of all terms in the vocabulary

Choosing the Naive Bayes classification method has many important benefits. First, Naive Bayes is a simple and computationally efficient algorithm, making it suitable for large datasets and large-sized data. In addition, Naive Bayes is highly scalable with many features and data points, and even with a relatively low amount of training data, it can still perform well. Naive Bayes' ability also handles unimportant features as long as they function independently. This advantage is particularly valuable in sentiment analysis as Naive Bayes models can handle large and complex text data efficiently. In addition, the predictability and speed of Naive Bayes training is essential for applications that require real-time analysis. Although the algorithm is simple, the results are often competitive compared to more complex methods, especially when combined with techniques such as TF-IDF.

The combination of other methods besides Naive Bayes in sentiment analysis can improve the accuracy and effectiveness of the model. For example, combining Naive Bayes with TF-IDF can make the words in the document have more relevant weights, which improves the model's ability to distinguish between important and unimportant words. In addition, ensemble methods and Naive Bayes can improve the accuracy of the model by reducing bias and variance. With the help of these methods, the model can capture various elements from the text data and produce better classification. In the classification stage, Naive Bayes is used to predict the label for each document in the data set, and its performance is evaluated using metrics such as F1 score, recall, accuracy, and precision. Sentiment analysis can be performed more effectively by using the advantages of Naive Bayes and these additional methods, resulting in more accurate classification results and more useful information from the text data.

E. Classification using of Logistic Regression

One kind of supervised machine learning is called logistic regression that can be used to analyze data and describe between one or more prediction variables and one response variable. The response variable from Logistic Regression only has a value between 0 and 1 so that it will produce positive and negative sentiment classes [7], with the limit between the two being a value of 0.5 [16]. The theory of the Logistic Regression method can be seen below:

$$g(X) = sigmoid(\alpha + \beta . X \tag{10}$$

$$signoid(x) = 1/(1 + exp - x) \tag{11}$$

Logistic Regression method formula (Source : [17])
Description:
Y : Response or dependent variable. This column indicates the sentiment of the review and contains manual labeling
α : Constant
β : Regression coefficient (slope); the amount of response caused by the predictor
X : A numerical number obtained from the transformation process of the pre-processed review text into a sparse matrix of values weighted by the importance of the token.

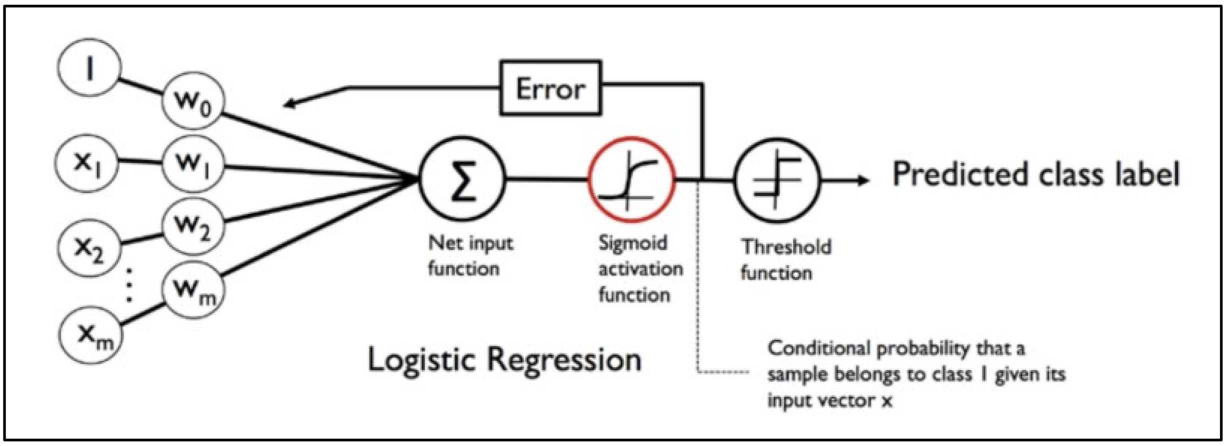

Fig. 2. Illustration Logistic Regression

Logistic regression is the most popular method for sentiment analysis because it is very practical, easy to use, and works well for binary classification problems such as determining whether a text has a positive or negative sentiment. It produces probabilistic results, which help us understand how likely a text is to belong to a particular sentiment class. This capability is very useful for sentiment analysis as it provides additional information about the confidence level of the model in its classification. By combining it with other techniques such as TF-IDF (Term Frequency-Inverse Document Frequency), logistic regression can easily improve the accuracy of sentiment analysis. TF-IDF assigns more relevant weights to words present in the document. This improves the model's ability to distinguish important words from unimportant ones in the text and capture meanings that are more appropriate to the context.

Moreover, combining logistic regression with other techniques such as Naive Bayes can significantly improve sentiment analysis results. Naive Bayes is effective in handling large-dimensional text data due to the assumption of independence between features, while logistic regression offers easy-to-understand probabilistic results. To improve model accuracy and reduce variance and bias, ensemble techniques such as bagging or boosting can also be applied. Sentiment analysis can be performed more effectively, providing more accurate classification results and more useful information from text data, by utilizing the respective advantages of these algorithms and techniques. In the classification stage, the label for each document in the data set is predicted using logistic regression. Furthermore, the results are assessed with metrics such as accuracy, precision, recall, F1 score, and its confusion matrix.

F. Lexicon

Lexicon-based approaches use sentiment lexicons that contain information about which words and phrases are positive and negative [3]. The Lexicon-Based Approach classifies a sentiment of each opinion, so that sentiment sentence data can be classified according to negative or positive classes [17]. Sentiment words, or words containing opinions, are the most significant indicators of sentiment. For example, the words "good", "very good", and "great" represent positive feelings, while the words "bad", "ugly", and "broken" represent negative feelings. Words and idioms, such as arms and legs, as well as unit words, are very important for studying sentiment[18]. VADER scores text from "(-4) Strongly Negative" to "(4) Strongly Positive", with an allowance of "(0) Neutral". To calculate the sentiment score of each word in a sentence registered in the VADER lexicon, the sentiment score of each word is summed. Each class will have a normalized polarity score between -1 indicating the most negative score and 1 indicating the most positive score, with a value of 0 indicating neutral. The compound score result shows the total sentiment score, where a value of -1 indicates the most negative score and a value of 1 indicates the most positive score. Hutto uses the following score normalization formula[18]:

$$\frac{x}{\sqrt{x^2 + a}} \tag{12}$$

where x is the sum of the sentiment values of all the words that make up the sentence, and alpha is the normalization constant. As a result, VADER sentiment analysis is more suitable for short documents such as tweets and sentences than for larger documents[18]. The advantage of lexicon-based methods lies in their ability to provide standardized sentiment scores based on predefined words and phrases. It also allows for a consistent and repeatable analysis process and quickly generates sentiment scoring for simple text data. However, this method has some drawbacks when dealing with complex situations or sensitive feelings, which are not covered by the current vocabulary. By combining lexicon-based methods with machine learning-based methods, sentiment analysis results can be improved. This is because lexicon-based methods can teach the model to understand more complex contexts and handle sentiment nuances that are not accommodated by the lexicon. By combining these two approaches, sentiment analysis can leverage the advantages of lexicon-based methods in terms of speed and ease of implementation, while utilizing machine learning to handle more complicated aspects of the text.

The VADER dictionary, a subset of the English dictionary [18], was used to identify sentiment polarity in this study due to its ability to provide a judgment close to human perception [19]. VADER is specifically designed for short texts such as social media comments and tweets. To guarantee the accuracy of the analysis with VADER, the googletrans library was used to translate the data from native language to English. However, the analysis results may be affected by the translation process due to possible loss of original meaning or nuance. After translation, sentiment scores were calculated based on each word present in the sentence, summed to produce a composite score, and then normalized and clustered. Values exceeding 0 are considered positive and values less than 0 are considered negative. While VADER works well for short, informative texts, it cannot handle more complex contexts, and translation errors may affect the analysis results. In addition, it may not account for all cultural variations and linguistic variations that can affect perception, especially in longer texts or in languages with different structures and expressions than English, which may not be included in this lexicon.

TABLE IV
LEXICON

Original Text	Compound Score	Sentiments
good	0,4404	Positive
upgrade g strange application now it's open	-0,2023	Negative
Basically Kredivo is really the best the limit is great thank you Kredivo good luck	0,9515	Positive
Kredivo give me a star please increase the limit don't just pay that much smoothly never be late I've upgraded the premium limit just don't increase it that much	0,4468	Positive
Boss Q please help me pay my bill which is verified. I paid my bill from yesterday. I also included proof that the payment transaction was also successful.	0,8126	Positive

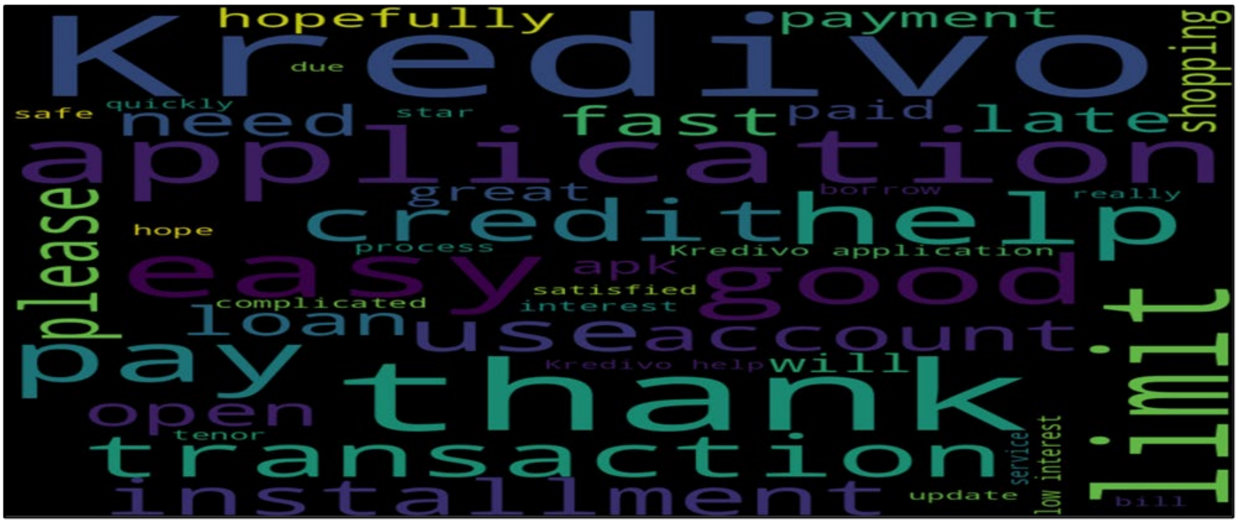

Fig. 3. Lexicon Positif

G. Model Voting

The Combination voting feature is a vote to determine positive and negative results carried out between the results of naïve bayes with multinomial bayes type, logistic regression, and lexicon. After the Lexicon stage is complete, then perform a combination by voting to determine positive and negative results, the determination comes from the results of Naive Bayes, Logistic Regression, and Lexicon. Determination of voting is done by the way if the results of Naive Bayes and Logistic Regression are the same then the label results are directly determined, but if the results of Naive Bayes and Logistic Regression are different then voting will be taken where the most votes will be taken between the results of Naive Bayes, Logistic Regression, and Lexicon.

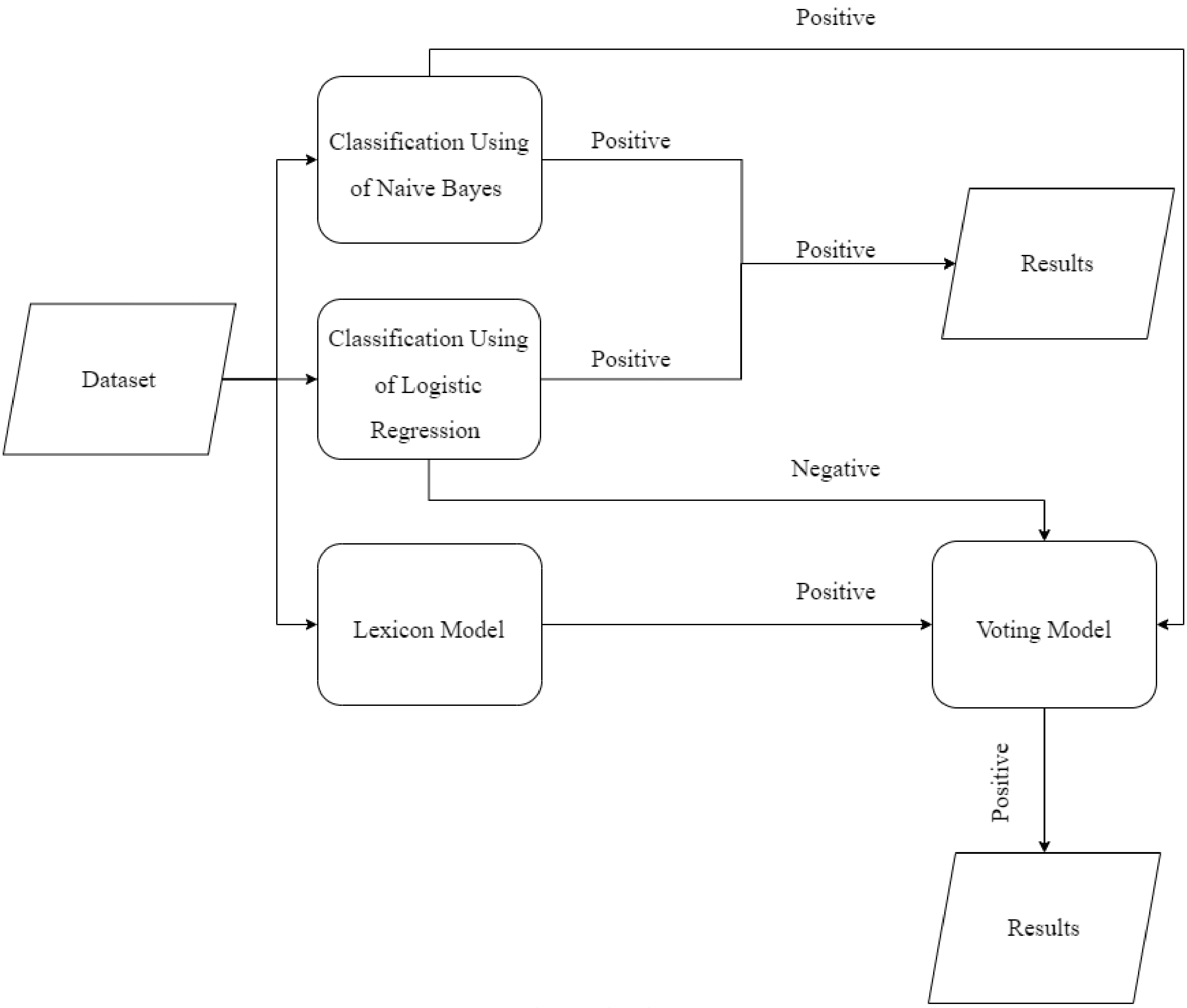

Fig. 4. Voting Flow

H. *Evaluation*

1. *K-fold Cross Validation*

   One method of cross validation is K-fold cross validation, which divides the data into k parts of equal size. Instructions and tests are performed k times[20].

2. *Confusion Matrix*

   Confusion matrix, which offers a thorough summary of the model's prediction outputs, is a crucial performance evaluation tool in classification data mining. The primary components of this matrix are False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN). Recall (sensitivity) measures the proportion of positives that are successfully identified, while precision measures how many positive predictions are actually correct. The model's total predicted success is measured by accuracy. Recall, precision, and accuracy can be calculated using specific formulas based on the elements of the confusion matrix.

TABLE V
CONFUSION MATRIX

Predicted Value	Actual Values	
	Postive(1)	Negative(0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Description:

- *True Positive (TP) where the actual value is 1 and the predicted value is 1*
- *False Positive (FP) where the actual value is 0 and the predicted value is 1*
- *False Negative (FN) where the actual value is 1 and the predicted value is 0*
- True Negative (TN) where the actual value is 0 and the predicted value is 0

Here are the formulas for the matrix that will be used.

1. Accuracy

   Accuracy is the ratio between the number of correct predictions and the overall data evaluated. The accuracy formula can be found in formula (12).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{13}$$

2. Precision

   Precision measures how accurate the positive predictions made by the model are by comparing them to the total positive predictions generated. The formula for precision can be found in formula (13).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

3. Recall

   Recall measures the extent to which the model successfully identifies data that is actually positive by comparing it to the total actual positive predictions. The formula for recall can be found in formula (14).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

4. F1-Score

   F1-Score is the harmonic mean between precision and recall, which is used to combine the two metrics into one evaluation measure that reflects overall performance. The F1-Score formula can be found in formula (15).

$$\text{F1} - \text{Score} = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right) \tag{16}$$

III. Result and Discussion

A. Result

Testing is done using the naïve bayes model, logistic regression model, voting model. utilizing a confusion matrix, 10 folds cross-validation, and an 80/20 split of training and test data. the performance results of the Naive Bayes model, Logistic Regression, and voting model based on confusion matrix can be seen in the Table and the results based on cross validation can be seen in the Table .

TABLE VI
RESULT CONFUSION MATRIX

Method	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	83.54%	83.38%	83.54%	83.08%
Logistic Regression	86.19%	86.13%	86.19%	86.16%
Voting	89.62%	89.61%	89.62%	89.62%

TABLE VII
RESULT CROSS VALIDATION

Fold	Naïve Bayes Accuracy	Logistic Regression Accuracy	Voting Accuracy
Fold 1	84.21%	87.40%	89.62%
Fold 2	80.23%	85.01%	89.62%
Fold 3	84.21%	86.87%	89.62%
Fold 4	82.89%	84.48%	89.62%

Fold 5	82.62%	85.14%	89.62%
Fold 6	83.93%	87.25%	89.62%
Fold 7	83.79%	88.18%	89.62%
Fold 8	82.73%	84.06%	89.62%
Fold 9	83.39%	84.32%	89.62%
Fold 10	83.00%	85.12%	89.62%
Best Acccuracy	84.21%	88.18%	89.62%
Average Accuracy	83.10%	85.78%	89.62%

B. Discussion

The results of this study compare the effectiveness of three classification methods: Voting, Logistic Regression, and Naive Bayes. With an average accuracy of 89.62%, the Voting method far outperformed Logistic Regression (85.78%) and Naive Bayes (83.10%) on every round of testing, showing that the Voting method is able to generate consistent and stable data. While Logistic Regression is effective in many situations with a linear relationship between features and classes, and Naive Bayes provides speed and simplicity under the assumption of independence between features, the Voting method mitigates the individual weaknesses of each model by reducing errors that may occur due to bias or inaccuracy of a single model. However, the Voting method can be more complex and requires more computational time, its effectiveness is highly dependent on the diversity of the models combined.

The results show that the F1-Score value is very useful for assessing the performance of different classification methods. F1-Score is a score that combines precision and recall into one number that shows the balance between the two. Based on the data collected, the Voting method has the highest F1-Score, which is 89.62%. This shows that this method is not only more accurate but also has a better balance between precision and recall. In other words, voting is consistently very effective in gathering positive information and reducing negative information. Nonetheless, Logistic Regression has an F1-Score of 86.16%, which is also quite good, but still below voting. Voting provides more stable and reliable classification performance compared to logistic regression and Naive Bayes as it offers the best balance between precision and gain. This difference leads to better performance of voting overall; it demonstrates its ability to better balance gain and precision. This is in line with previous research which also shows that voting methods often outperform single models for classification tasks. The effectiveness of voting is highly dependent on the diversity and quality of the models used. While Logistic Regression and Naive Bayes each have their strengths and weaknesses, the combined advantages of Voting methods help to overcome their respective shortcomings, resulting in more consistent and reliable performance.

Previous studies on sentiment analysis show that the accuracy rate and F1 Score are different for various algorithms. For comparison, the K-Nearest Neighbor (KNN) and Naïve Bayes Classifier (NBC) algorithms showed an accuracy of 79.67% and 78.86% on sentiment analysis of PPKM policies during the COVID-19 pandemic. The MyPertamina app, available on Google Playstore, uses the Naïve Bayes Classifier to provide sentiment analysis of online loans with 87% accuracy, 86% precision, and 90% recall. With an 80:20 data split for training and testing, sentiment analysis on Twitter using Logistic Regression found an accuracy of 78.57%, precision of 76.92%, recall of 83.3%, and F1-Score of 80%. Sentiment analysis of mobile telecommunication services on Twitter using SVM shows 79% accuracy, 65% precision, 97% recall, and 55.34% F1-Score. On the other hand, the evaluation of digital population applications shows Logistic Regression has an accuracy of 78.83%, recall of 55.84%, precision of 71.63%, and F1-Score of 55.34%. The Voting method can overcome its shortcomings and produce more stable and reliable performance by combining the advantages of each model. These advantages are in line with the findings of previous studies which show that combining models such as the Voting method can result in significant improvements in data classification quality.

Specifically, the results of this study show that the Voting method has an accuracy of 89.62%, higher than the highest accuracy of 87% achieved by the Naïve Bayes Classifier in the previous study. In addition, the Voting method has an F1-Score of 89.62%, surpassing the recall of 90% and precision of 86% achieved by the Naïve Bayes Classifier study on the MyPertamina application. The Voting method shows consistent and superior performance in various areas, with an accuracy of 89.62%-89.62%. This shows the strength of this method. This consistency is contrary to the variations found in previous studies. For example, Naive Bayes accuracy ranges from 75% to 87%, and logistic regression ranges from 78.57% to 78.83%. The voting method was able to leverage the strengths of various models and produce superior and stable results across multiple data sets, reaffirming the benefits of model building in achieving high-performance classification.

IV. Conclusion

The need to improve classification accuracy in sentiment analysis of online loan applications drives this research. To overcome the limitations of a single classification model, we suggest using a selection method. This process combines the prediction results from three models-Naïve Bayes, Logistic Regression, and Lexicon-based analysis. The final decision is based on the majority vote of the three models, and Lexicon serves as the decider in case of discrepancies between Naïve Bayes and Logistic Regression.

Based on the 10 folds cross-validation results, Naive Bayes achieved an average accuracy of 83.10% and Logistic Regression achieved an average accuracy of 85.78%. Both of these results can be considered quite good, but there is still room for improvement. The voting method showed very satisfactory results, with an average accuracy of 89.62%. Even more impressively, this accuracy was consistent across all folds, showing high consistency. The effectiveness of the voting method in improving classification performance is demonstrated by the 3.84% increase in accuracy over Logistic Regression and 6.52% over Naive Bayes. These results show that group methods such as voting can effectively combine the strengths of various classification models, resulting in more accurate and robust predictions.

References

[1] M. I. Ghozali, W. H. Sugiharto, and A. Fajar Iskandar, "KLIK: Kajian Ilmiah Informatika dan Komputer Analisis Sentimen Pinjaman Online Di Media Sosial Twitter Menggunakan Metode Naive Bayes," *Media Online)*, vol. 3, no. 6, pp. 1340–1348, 2023, doi: 10.30865/klik.v3i6.936.
[2] M. K. Elhadad, K. F. Li, and F. Gebali, "Detecting misleading information on COVID-19," *IEEE Access*, vol. 8, pp. 165201–165215, 2020, doi: 10.1109/ACCESS.2020.3022867.
[3] "DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN METODE LEVENSHTEIN DISTANCE SKRIPSI."
[4] F. Romadoni, Y. Umaidah, and B. N. Sari, "Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 2, pp. 247–253, Jul. 2020, doi: 10.32736/sisfokom.v9i2.903.
[5] A. V. T. R. Rihan Maulana, "ANALISIS SENTIMEN ULASAN APLIKASI MYPERTAMINA PADA GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA NBC," *Jurnal Teknologi Terpadu*, vol. 9, no. 1, pp. 42–48, Jul. 2023, doi: https://doi.org/10.54914/jtt.v9i1.609.
[6] B. Ramdani, A. Dwi Saputra, M. Rafli Alta Zahir, I. Komputer, and U. Bhayangkara Jakarta Raya, "Analisis Sentimen Terhadap Ulasan Aplikasi Pinjaman Online (PINJOL) di Google Play Store Menggunakan Naive Baiyes Classifer," 2023.
[7] S. A. Assaidi and F. Amin, "Analisis Sentimen Evaluasi Pembelajaran Tatap Muka 100 Persen pada Pengguna Twitter menggunakan Metode Logistic Regression."
[8] B. Setiawan, K. Ahmad Baihaqi, E. Nurlaelasari, and H. Hikmayanti Handayani, "Analisis Sentimen Ulasan Aplikasi Identitas Kependudukan Digital Menggunakan Algoritma Logistic Regression dan K-Nearest Neighbor," *Technology and Science (BITS)*, vol. 6, no. 1, pp. 533–540, 2024, doi: 10.47065/bits.v6i1.5389.
[9] S. S. Salim and J. Mayary, "ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP DOMPET ELEKTRONIK DENGAN METODE LEXICON BASED DAN K – NEAREST NEIGHBOR," *Jurnal Ilmiah Informatika Komputer*, vol. 25, no. 1, pp. 1–17, 2020, doi: 10.35760/ik.2020.v25i1.2411.
[10] I. L. Kharisma, D. A. Septiani, A. Fergina, and K. Kamdan, "Penerapan Algoritma Decision Tree untuk Ulasan Aplikasi Vidio di Google Play," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, no. 2, pp. 218–226, Sep. 2023, doi: 10.25077/teknosi.v9i2.2023.218-226.
[11] Indra, Agus Umar Hamdani, Suci Setiawati, Zena Dwi Mentari, and Mauridhy Hery Purnomo, "Comparison of K-NN, SVM, and Random Forest Algorithm for Detecting Hoax on Indonesian Election 2024," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 13, no. 1, Mar. 2024, doi: 10.23887/janapati.v13i1.76079.
[12] N. Komang *et al.*, "Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia," *MERPATI*, vol. 6, no. 2, 2018.
[13] M. Farid Rifai, H. Jatnika, B. Valentino, and S. Tinggi Teknik PLN, "Penerapan Algoritma Naïve Bayes Pada Sistem Prediksi Tingkat Kelulusan Peserta Sertifikasi Microsoft Office Specialist (MOS)," vol. 12, no. 2, 2019.
[14] A. Sabrani, I. W. Gede Putu Wirarama Wedashwara, and F. Bimantoro, "METODE MULTINOMIAL NAÏVE BAYES UNTUK KLASIFIKASI ARTIKEL ONLINE TENTANG GEMPA DI INDONESIA (Multinomial Naïve Bayes Method for Classification of Online Article About Earthquake in Indonesia)." [Online]. Available: http://jtika.if.unram.ac.id/index.php/JTIKA/
[15] A. Rahman and A. Doewes, "Online News Classification Using Multinomial Naive Bayes." [Online]. Available: www.kompas.com
[16] X. Wan, "The influence of polynomial order in logistic regression on decision boundary," in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics Publishing, Jun. 2019. doi: 10.1088/1755-1315/267/4/042077.
[17] Ash Shiddicky and Surya Agustian, "Analisis Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 pada Media Sosial Twitter menggunakan Metode Logistic Regression," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 2, pp. 99–106, Aug. 2022, doi: 10.37859/coscitech.v3i2.3836.
[18] D. Abimanyu, E. Budianita, E. Pandu Cynthia, F. Yanto, P. Studi Teknik Informatika, and F. Sains Dan Teknologi, "Analisis Sentimen Akun Twitter Apex Legends Menggunakan VADER," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 5, no. 3, 2022, [Online]. Available: https://techno.kompas.com
[19] J. Garay, R. Yap, and M. J. Sabellano, "An analysis on the insights of the anti-vaccine movement from social media posts using k-means clustering algorithm and VADER sentiment analyzer," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Mar. 2019. doi: 10.1088/1757-899X/482/1/012043.
[20] M. Farras,) Viny, C. Mawardi, and T. Sutrisno, "Aplikasi Analisis Sentimen Komentar Pengguna Genshin Impact Di Play Store."