Korespondensi Journal of Innovation Information Technology and Application (**JINITA**)

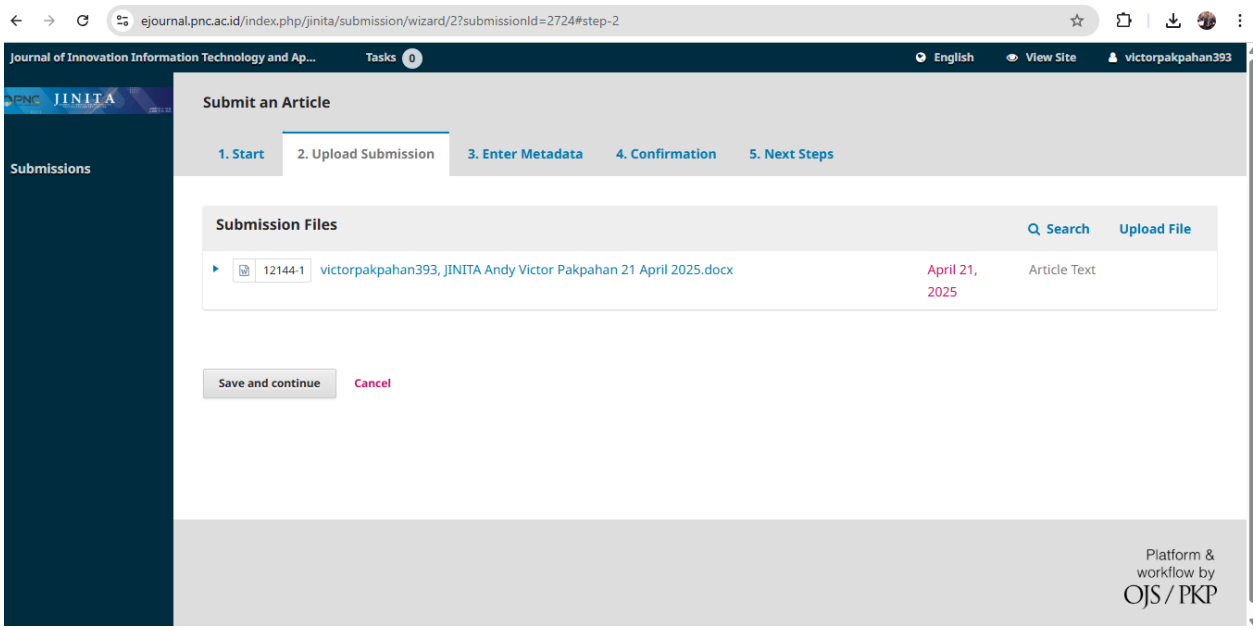
Sentiment Analysis Using Stacking Ensemble After the 2024 Indonesian Election Results

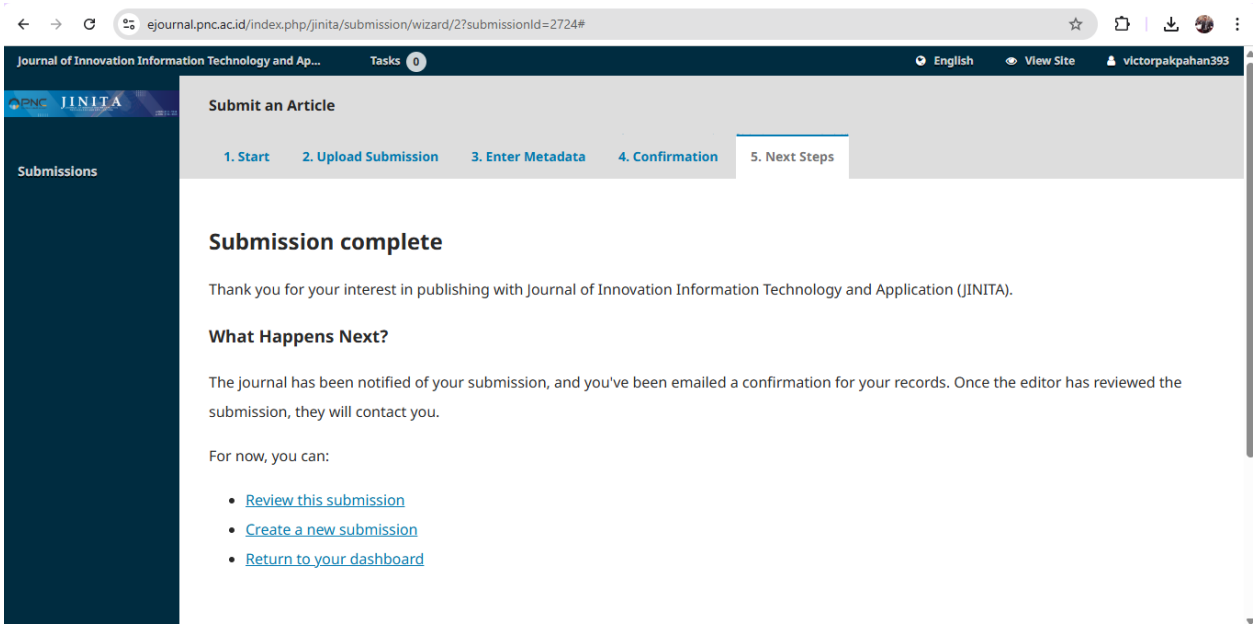
Vol. 7 No. 1 (2025): JINITA, June 2025

DOI: https://doi.org/10.35970/jinita.v7i1

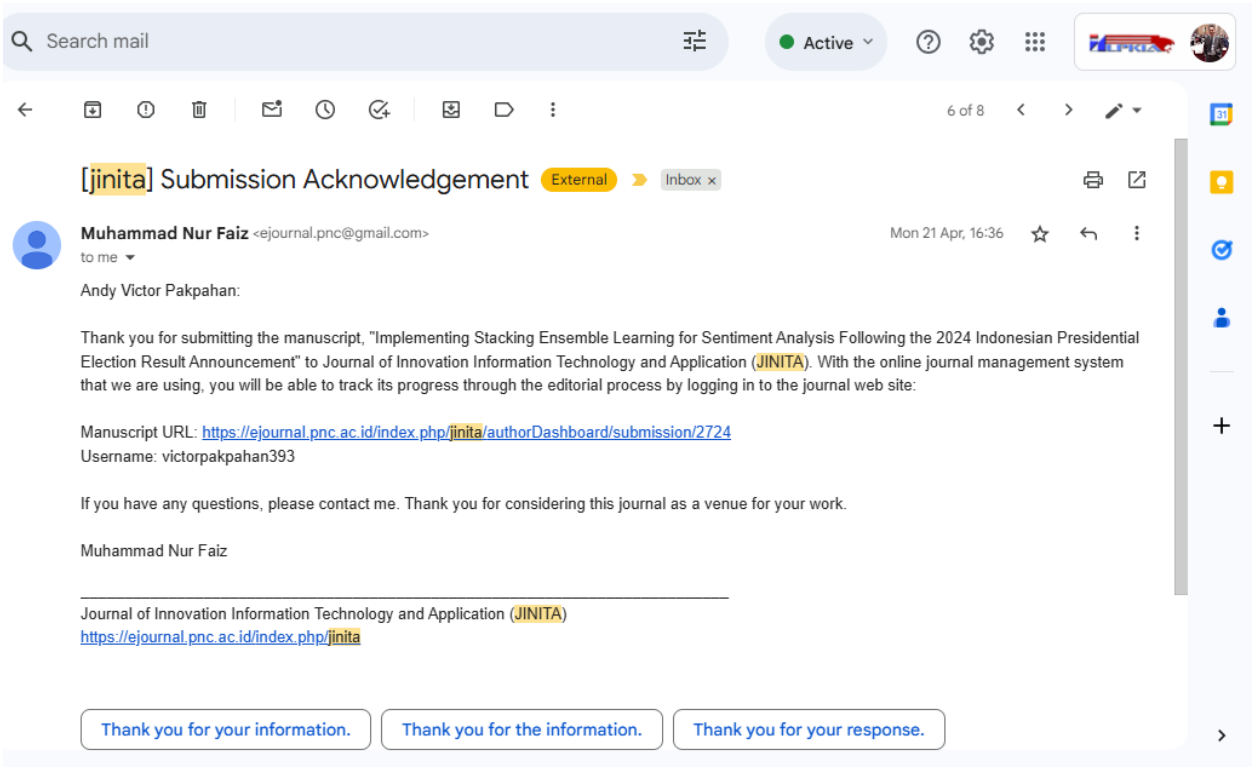
No	Tanggal	Kegiatan
1	21 April 2025	Unggah Artikel ke JINITA
2	21 April 2025	JINITA Submission Acknowledgement
3	7 Mei 2025	JINITA Article Format Adjusment
4	31 Mei 2025	JINITA Editor Decision – Revisions Required
5	2 Jun 2025	JINITA Editor Decision – Revisions Required - Reminder
6	5 Juni 2025	Unggah Revisi Artikel
7	25 Juni 2025	JINITA Editor Decision – Accepted Submission – Letter of Acceptance (LoA)
8	30 Juni 2025	Publikasi Paper https://ejournal.pnc.ac.id/index.php/jinita/issue/current

1. Unggah Artikel ke JINITA

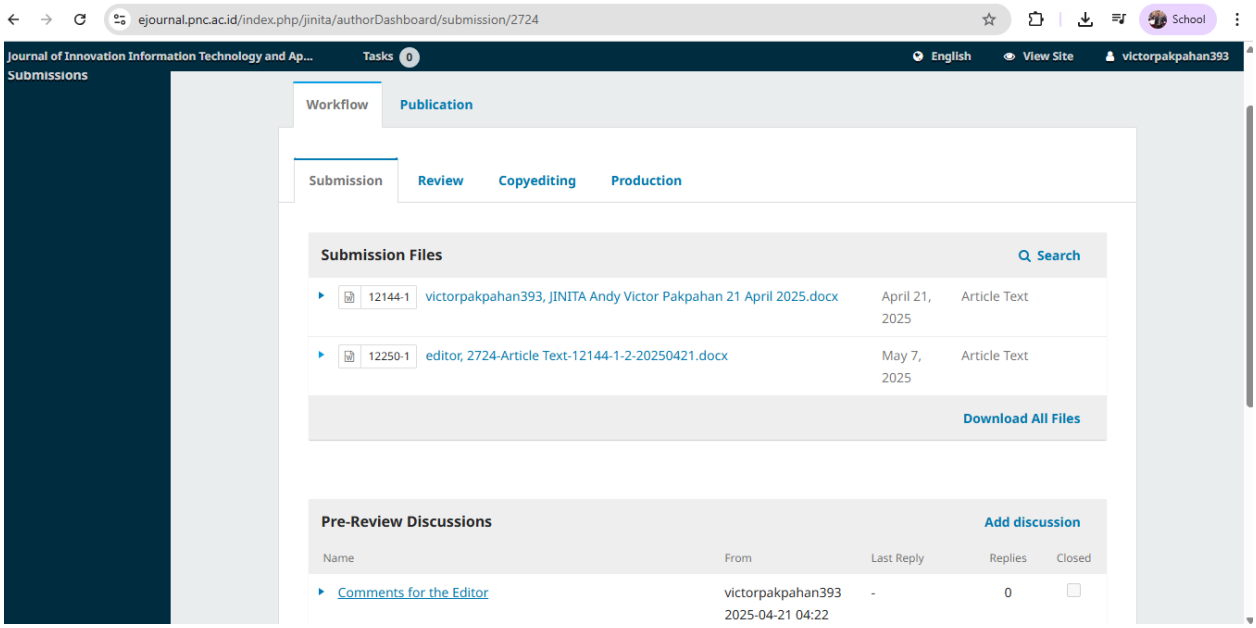




2. JINITA Submission Acknowledgement



3. JINITA Article Format Adjusment



Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Implementing Stacking Ensemble Learning for Sentiment Analysis Following the 2024 Indonesian Presidential Election Result Announcement

ARTICLE INFO

Article history:
Received 24 January 2020
Revised 30 April 2020
Accepted 2 December 2020
Available online xxx

Keywords:
Ensemble Learning,
Stacking,
Sentiment Analysis,
Data Mining,
Machine Learning

IEEE style in citing this article:
Jinita and J. Jinita, "Article Title," *Journal of Innovation Information Technology and Application*, vol. 4, no. 1, pp. 1-10, 2022. [Fill citation heading]

ABSTRACT

Sentiment analysis is a technique for processing text with the aim of identifying opinions and emotions within a sentence. In its approach, machine learning has become a commonly used method. Several algorithms such as Naive Bayes, Support Vector Machine (SVM), and Random Forest are often used in this analysis. However, creating a model that has optimal accuracy is still a challenge, especially for sentiment analysis of unstructured text data. The aim of this research is to try to improve the accuracy of the sentiment analysis model by using the ensemble learning stacking method approach, namely a method for combining several base models to produce better model performance. The algorithms that will be the base models in this stacking are Random Forest, Naive Bayes and SVM, while the model that makes the final prediction or what is called a meta model will use Random Forest. This sentiment analysis was carried out in a specific context, namely public opinion after the announcement of the results of the 2024 Indonesian presidential election. The dataset analyzed using reviews or public opinion regarding the results of the 2024 Indonesian presidential election was 6737 tweet data collected via the X platform using web scraping. As a result, the Naïve Bayes method got an accuracy of 66.84%, SVM got an accuracy of 77.74%, Random Forest got an accuracy of 74.78% and stacking got an accuracy of 81.53%.

1. INTRODUCTION

The Ensemble Learning is a method in machine learning that combines several models to create a new model that is stronger than and has superior performance compared to when the algorithms are used individually [1], [2]. There are several ensemble learning techniques such as bagging, stacking, averaging and boosting, each technique is distinguished by how the model is trained and combined [1].

Stacking is an ensemble learning technique that works by combining the results of several different base-models. Each base-model will learn and have its own prediction results, after that a final model will be created which will combine the prediction results of all the base-models which is called a meta-model [3], [4]. The Stacking technique is based on the idea that each basic model has its own advantages and disadvantages [5]. By combining predictions from different base-models, the resulting meta-model can learn and balance these advantages and disadvantages appropriately, so that the overall performance of the stacking model can exceed the performance of any individual model and makes it a fairly good technique for improving predictive power of the classifier [6], [7]. This is the advantage of the stacking technique compared to other ensemble learning techniques and makes stacking a suitable technique for creating models for processing quite complex data such as sentiment analysis [8]

Sentiment analysis is the process of understanding, extracting and processing textual data automatically to obtain information on opinions, feelings and emotions contained in a sentence [9]. Sentiment analysis aims to understand a person's level of satisfaction and dissatisfaction with a service or product, as well as understanding public perceptions regarding a person's agreement and disagreement with a particular topic [10]

*) Corresponding Author : abang@lpkia.ac.id

Sentiment analysis is generally made using classification algorithm models such as Support Vector Machine (SVM), Decission Tree, K-Nearest-Neighbor (KNN), Naïve Bayes, Random Forest, etc [11], [12]. Several classification algorithms have been used in several previous studies regarding sentiment analysis carried out on opinions taken from social media X or Twitter in Indonesian and each algorithm has different accuracy [11]. Compared the SVM algorithm with other algorithms such as Naïve Bayes, Decission Tree and KNN in sentiment analysis with different cases or topics, the results is that SVM accuracy is better when compared to other algorithms. Even though Naïve Bayes is not superior in accuracy to the SVM algorithm, if we refer to research conducted Naïve Bayes still has better accuracy results when compared to the Decision Tree and KNN algorithms [13]. Then, if we refer to research which compared the Random Forest algorithm with other algorithms such as Naïve Bayes, KNN, Decission Tree and Logistic Regression, it can be seen that Random Forest produces better accuracy than other algorithms including SVM [14]. Other research that shows that Random Forest is superior to SVM is research From these studies, it can be seen that the Random Forest, SVM and Naïve Bayes algorithms are some of the algorithms with the best accuracy in terms of sentiment analysis.

Even so, sentiment analysis is not an easy task to do. The complexity of language and variations in human expressions in various sentences make sentiment analysis a challenge [15], [16]. Building a model that can produce accuracy and good performance is also a challenge in sentiment analysis [17] especially sentiment analysis of unstructured text, for example data taken from social media such as X or Twitter has its own challenges because the language used is usually not appropriate. standard words, involving abbreviations, as well as words that are not in the dictionary, thus affecting accuracy[18], [19]. So the accuracy of the sentiment analysis model can still be improved with the help of other methods, for example by using the ensemble stacking method.

Based on the description above, a sentiment analysis model will be built using the ensemble learning stacking method, with the aim of increasing the accuracy of the model in sentiment analysis on unstructured text, which in this case is data collected via the social media platform [10], [20]. This sentiment is the public's opinion regarding the results of the 2024 Indonesian Presidential Election. We propose to use Naive Bayes, Random Forest and Support Vector Machine as base models and Random Forest as a meta model because these models are suitable for sentiment analysis and have been widely used by previous researchers. Beside from that, these models also have different characteristics.

2. METHOD

The research flow presented in this experiment outlines the structured steps taken to achieve the objectives of the study. It begins with the identification of the problem, which serves as the foundation for formulating the research questions and determining the appropriate methodology. This initial phase is crucial to ensure that the research direction is clear and aligned with the intended goals.

Following problem identification, the flow continues through stages such as data collection, analysis, and interpretation. Each step is interconnected, allowing the process to build logically upon the previous one. This structured approach not only helps in maintaining the consistency of the study but also enhances the reliability and validity of the results obtained.

Figure 1 provides a visual representation of this research flow. It serves as a guide to understand how the experiment was conducted from start to finish. By presenting the process in a flowchart format, it becomes easier to grasp the overall methodology and appreciate the systematic effort involved in reaching the research conclusions.

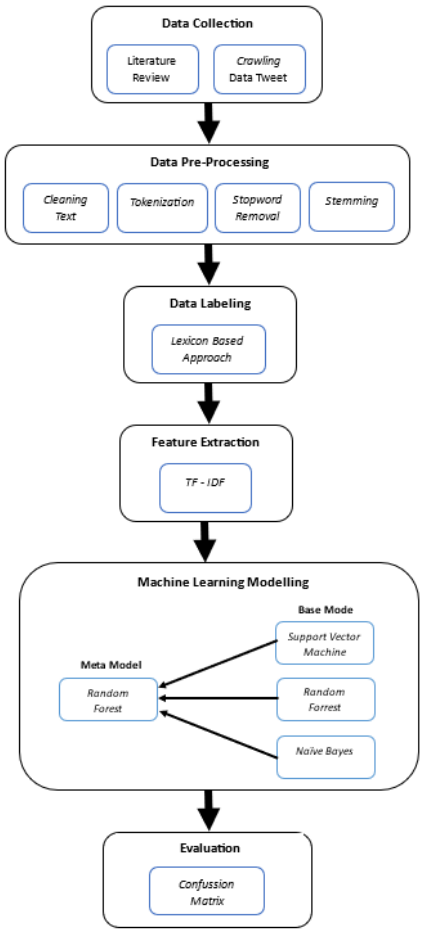


Fig 1. Flow of Research

A. DATA COLLECTION

Based on the description above, a sentiment analysis model will be built using the ensemble learning stacking method, with the aim of increasing

B. DATA PRE-PROCESSING

This process includes a series of steps to prepare the data before creating a sentiment analysis model. Stages that will be carried out in the process pre-processing data is as follows:

1) CLEANING TEXT

At this stage, text data will be cleaned has been collected from scrapping results so that the text can be made easier processed by the next stage. Data cleaning includes several processes such as deleting numbers and symbols, changing text to lowercase and also normalize the text or change each word in a sentence becomes standard or normal form for omission non-standard words, abbreviations, slang words, typo words etc. Text normalization This is done by referring to the dictionary provided which contains non-standard words and actual standard words.

2) TOKENIZATION

At this stage, every text data has been cleaned will be convert into small parts of each word in sentences are called tokens. For example, sentence “Indonesia Lebih Maju” will be converted to ["Indonesia", "lebih", "maju"]

3) STOPWORD REMOVAL

At this stage, any common words are not make significant contributions to the meaning of the text will be removed. The stopword dictionary will be taken from a library that has provided a list the stop words are

Sastrawi. Some examples of words included in the stopword and will be deleted are like “yang”, “dan”, “di”, “adalah”.

4) STEMMING

At this stage, every word in the text will be changed be the basic word. Words with the same ending or words those with affixes will be changed to the basic form.

C. DATA LABELING

The method that will be used for data labeling is Lexicon Based. Lexicon based Approach can be used to create labeled training datasets for sentiment analysis machine learning algorithms that require labels at the start his training [23]. The idea behind the lexicon based approach is that the meaning of a text is greatly influenced by the polarity of the words and phrases in inside. This includes words such as adjectives, adverbs, nouns, verbs, as well as phrases and sentences that contain them [24]. This approach makes use of a dictionary or list of words with predefined sentiment labels. Any data will be carried out Check the total score of positive words and negative words. If the word score positive exceeds negative scores, then the label is positive, and vice versa then the label is negative. However, if the score is the same or 0, then the label is will be neutral.

D. FEATURE EXTRACTION

In this process, feature extraction will be carried out using the Term Frequency-Inverse Document Frequency (TF-IDF). At this stage, every tweet will be represented as a numerical feature vector, where each component The vector will represent the weight of each word in the existing word dictionary. This weight calculated based on the frequency of occurrence of words in tweets (TF) and inverse proportional to the occurrence of the word in the entire collection of tweets (IDF). This feature extraction process aims to change the tweet text into a numerical representation that can be used by the model to perform analysis further sentiment. The formulas of TF-IDF are as follows:

$$TF(t,d) = \frac{number\ of\ occurences\ of\ word\ in\ document}{total\ number\ if\ words\ in\ document} \quad (1)$$

$$IDF(t,D) = \log\left(\frac{N}{df(t,D)}\right) \quad (2)$$

$$TF - IDF(t,d) = TF(t,d)x\ IDF(t,D) \quad (3)$$

- N : Total number of documents in the collection
- df(t,D): Number of documents in the collection containing term t
- TF(t,d): Term Frequency of term t in document d
- IDF(t,D): Inverse Document Frequency of term t in all documents D
- W(t,d): Weight of term t in a document

E. STACKING MODELLING

At this stage a model will be built for sentiment analysis. Before that, The data will first be split into 2 parts, namely training data and testing data with a percentage of 80% training data and 20% test data. Then, it will be done training on each base model, namely naïve Bayes, support vector machine and random forest use data that has been split. Every base the model will generate predictions based on these features. Then as in the ensemble learning stacking technique stage, the output from the base model will be used as a feature for the meta model model, which is in charge combining predictions from the base model to produce a prediction the new one. The following is an illustration of the stacking model that will be combined.

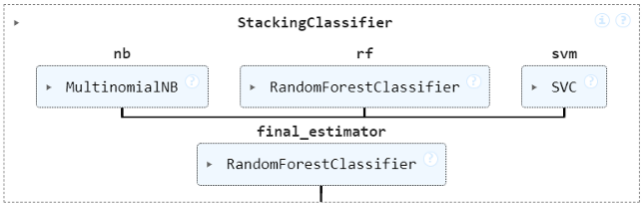


Fig 2. Stacking Model Illustration

Here is the algorithm for the stacking model that will be created

Table 1. Algortihm Stacking

Algorithm 1. Stacking

Input: X_train, y_train, x_test, base_models, meta_model

Ouput: prediction meta model

```
START
base_model_outputs_train = []
FOR model in base_models THEN
    model.fit(X_train, y_train)
    probas_train = model.predict_proba(X_train)
    base_model_outputs_train.append(probas_train)
END FOR
meta_features_train = np.hstack(base_model_outputs_train)
meta_model.fit(meta_features_train, y_train)
base_model_outputs_test = []
FOR model in base_models THEN
    probas_test = model.predict_proba(X_test)
    base_model_outputs_test.append(probas_test)
END FOR
meta_features_test = np.hstack(base_model_outputs_test)
final_predictions = meta_model.predict(meta_features_test)
END
```

F. EVALUATION

After the model building process is complete, the next step is evaluate model performance using confusion matrix. Evaluation is carried out against each base model and meta model itself, so you can see the comparison of classification results between models. Because this sentiment analysis has 3 labels, namely positive, negative and neutral (multi class), the calculation of each metric will use the concept of a weighted average where the formula is as follows

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} \quad (4)$$

$$Precission_{weighted} = \frac{\sum_{i=1}^{N}(Precission_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (5)$$

$$Recall_{weighted} = \frac{\sum_{i=1}^{N}(Recall_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (6)$$

$$F-1Score_{weighted} = \frac{\sum_{i=1}^{N}(F-1score_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (7)$$

These evaluation metrics provide a comprehensive view of the model's ability to correctly classify sentiments across all classes. By using weighted averages, the metrics take into account the proportion of each class, ensuring that imbalanced class distributions do not bias the results. This is particularly important in multiclass classification problems where some classes may dominate. The use of these formulas allows for a fair comparison of performance between base models and the meta model.

3. RESULTS AND DISCUSSION

A. WEB SCRAPPING RESULT

The total data collected was 8094 tweets with details of each keyword are as follows.

Table 2. Data Collected by keyword.

Keyword	Query Search	Result
Hasil pemilu	Hasil pemilu lang:id until:2024-04-30 since:2024-03-20 -filter:links -filter:repliess	3482 tweet
Hasil pemilu presiden	Hasil pemilu presiden lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	156 tweet
Hasil pilpres	Hasil pilpres lang:id until:2024-04-30 sinc e:2024-03-20 -filter:links -filter:repliess	1983 tweet
Pemenang pemilu	Pemenang pemilu lang:id until:2024-04-3 0 since:2024-03-20 -filter:links -filter:rep liess	1000 tweet
Pemenang pilpres	Pemenang pilpres lang:id until:2024-04-3 0 since:2024-03-20 -filter:links -filter:rep liess	808 tweet
Pemenang presiden	Pemenang presiden lang:id until:2024-04 30 since:2024-03-20 -filter:links -filter:re pliess	274 tweet
Pengumuman pemilu	Pengumuman pemilu lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	221 tweet
Pengumuman pilpres	Pengumuman pilpres lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	170 tweet

B. DATA PRE-PROCESSING

The pre-processing stage is the first step in preparing the dataset by carrying out several stages, namely cleaning text, tokenization, stopword removal and stemming as well as deleting duplicate data. Data cleaning of scrapped tweet text includes several processes such as deleting mentions, deleting hashtags, deleting retweets, deleting URLs, deleting non-alphanumeric characters, deleting double spaces and transform the text into lowercase. Then, text normalization will be carried out to change words such as abbreviations, non-standard words, and slang words into normal and formal words. Finally, to avoid data duplication, tweet data that has the same or duplicate sentences will be deleted. So the final total of tweet data that will be used in the next stage until the end is 6737 tweets. Here are the results

	lowered_text	normalized_text
1440	sudah tdk kaget dgn hasil pilpres yg di umumkan kpu malam ini sdh dr awal sejak mk dan kpu meloloskan gibram menjadi calon wakil prabowo menjadi salala satu kandidat capres pemilu bak sinetron yg kita sdh tau di mna ujung ceritanya mari kita tunggu sinetron episode berikut nya	sudah tidak kaget dengan hasil pilpres yang di umumkan kpu malam ini sudah dari awal sejak mk dan kpu meloloskan gibran menjadi calon wakil prabowo menjadi salah satu kandidat capres pemilu bak sinetron yang kita sudah tau di mana ujung ceritanya mari kita tunggu sinetron episode berikut nya
1441	pemilu hasil bansos 465 t	pemilu hasil bansos 465 t
1442	pemilu kok dianggap perang kacau ente ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok nte tidak cepat cuci muka sono biar siuman	pemilu kok dianggap perang kacau anda ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok anda tidak cepat cuci muka sono biar siuman
1443	tetap lawan amp tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yg tak beradab	tetap lawan sampai tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yang tidak beradab
1444	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc
1446	kok ngumumin hasil pemilu yg amat penting tengah malem bangett dahh orangorang juga rata2 udh pada tidur abis tarwih	kok ngumumin hasil pemilu yang amat penting tengah malem bangett dahh orangorang juga rata2 sudah pada tidur abis tarwih
1447	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yg berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yang berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont

Fig 3. Cleaning Text Result

Next, the tweet text that has been cleaned will be broken down into pieces of words in sentences called tokens. Following are some results from the tokenization process

	normalized_text	tokenized_text
1440	sudah tidak kaget dengan hasil pilpres yang di umumkan kpu malam ini sudah dari awal sejak mk dan kpu meloloskan gibran menjadi calon wakil prabowo menjadi salah satu kandidat capres pemilu bak sinetron yang kita sudah tau di mana ujung ceritanya mari kita tunggu sinetron episode berikut nya	['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini', 'sudah', 'dari', 'awal', 'sejak', 'mk', 'dan', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'yang', 'kita', 'sudah', 'tau', 'di', 'mana', 'ujung', 'ceritanya', 'mari', 'kita', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']
1441	pemilu hasil bansos 465 t	['pemilu', 'hasil', 'bansos', '465', 't']
1442	pemilu kok dianggap perang kacau anda ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok anda tidak cepat cuci muka sono biar siuman	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'anda', 'ini', 'ketum', 'nasdem', 'saja', 'sebagai', 'partai', 'penyokong', 'utama', 'paslon', 'no', '1', 'sudah', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'anda', 'tidak', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']
1443	tetap lawan sampai tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yang tidak beradab	['tetap', 'lawan', 'sampai', 'tolak', 'pemilu', 'curang', 'tidak', 'mengakui', 'pemimpin', 'dari', 'hasil', 'kecurangan', 'yang', 'tidak', 'beradab']
1444	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc	['saya', 'minta', 'kebijaksanaan', 'mui', 'yang', 'saat', 'ini', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', '2024', 'sebagai', 'umat', 'islam', 'kita', 'dituntut', 'untuk', 'adil', 'cc']
1446	kok ngumumin hasil pemilu yang amat penting tengah malem bangett dahh orangorang juga rata2 sudah pada tidur setelah tarwih	['kok', 'ngumumin', 'hasil', 'pemilu', 'yang', 'amat', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'juga', 'rata2', 'sudah', 'pada', 'tidur', 'setelah', 'tarwih']
1447	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yang berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont	['dari', 'hasil', 'pemilu', 'kali', 'ini', 'kita', 'mengetahui', 'bahwa', 'mayoritas', 'penduduk', 'indonesia', '1', 'minim', 'literasi', '2', 'tidak', 'suka', 'perdebatan', 'yang', 'berdasarkan', 'teori', 'data', 'dan', 'fakta', '3', 'cenderung', 'duniawi', '4', 'sekarang', 'ya', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lagi', '5', 'lebih', 'suka', 'kekerasan', 'cont']

Fig 4. Tokenization Result

Next, we enter the stopword removal stage, the tweet text which is already in token form will be removed by words that have no meaning in the text. This process was assisted with library assistance from Sastrawi. This library provides a function that can provide a list of words in Indonesian that have no meaning. The stopword removal process will refer to the list of words provided by the library. Here are the results

	tokenized_text	text_after_stopword
1440	['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini', 'sudah', 'dari', 'awal', 'sejak', 'mk', 'dan', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'yang', 'kita', 'sudah', 'tau', 'di', 'mana', 'ujung', 'ceritanya', 'mari', 'kita', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']	['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam', 'awal', 'sejak', 'mk', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'tau', 'mana', 'ujung', 'ceritanya', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']
1441	['pemilu', 'hasil', 'bansos', 't']	['pemilu', 'hasil', 'bansos', 't']
1442	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'anda', 'ini', 'ketum', 'nasdem', 'saja', 'sebagai', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'sudah', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'anda', 'tidak', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'ketum', 'nasdem', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']
1443	['tetap', 'lawan', 'sampai', 'tolak', 'pemilu', 'curang', 'tidak', 'mengakui', 'pemimpin', 'dari', 'hasil', 'kecurangan', 'yang', 'tidak', 'beradab']	['tetap', 'lawan', 'tolak', 'pemilu', 'curang', 'mengakui', 'pemimpin', 'hasil', 'kecurangan', 'beradab']
1444	['saya', 'minta', 'kebijaksanaan', 'mui', 'yang', 'saat', 'ini', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'sebagai', 'umat', 'islam', 'kita', 'dituntut', 'untuk', 'adil', 'cc']	['minta', 'kebijaksanaan', 'mui', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'umat', 'islam', 'dituntut', 'adil', 'cc']
1446	['kok', 'ngumumin', 'hasil', 'pemilu', 'yang', 'amat', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'juga', 'rata', 'sudah', 'pada', 'tidur', 'setelah', 'tarwih']	['kok', 'ngumumin', 'hasil', 'pemilu', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'rata', 'tidur', 'tarwih']
1447	['dari', 'hasil', 'pemilu', 'kali', 'ini', 'kita', 'mengetahui', 'bahwa', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'tidak', 'suka', 'perdebatan', 'yang', 'berdasarkan', 'teori', 'data', 'dan', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'ya', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lagi', 'lebih', 'suka', 'kekerasan', 'cont']	['hasil', 'pemilu', 'kali', 'mengetahui', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'suka', 'perdebatan', 'berdasarkan', 'teori', 'data', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lebih', 'suka', 'kekerasan', 'cont']

Fig 5. Stopword removal Result

The final process is stemming. This stage functions to change words that have affixes into basic words. This process also uses the stemming function that already exists in the Sastrawi library. In this process, the text form will be returned from token to plain text. The following are some texts that have undergone the stemming process

	text_after_stopword	text_after_stemming
1440	['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam', 'awal', 'sejak', 'mk', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'tau', 'mana', 'ujung', 'ceritanya', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']	kaget hasil pilpres umum kpu malam awal sejak mk kpu lolos gibran jadi calon wakil prabowo jadi salah satu kandidat capres pemilu bak sinetron tau mana ujung cerita tunggu sinetron episode ikut nya
1441	['pemilu', 'hasil', 'bansos', 't']	pemilu hasil bansos t
1442	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'ketum', 'nasdem', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']	pemilu kok anggap perang kacau tum nasdem partai sokong utama paslon no nyata terima hasil pemilu mosok cepat cuci muka sono biar siuman
1443	['tetap', 'lawan', 'tolak', 'pemilu', 'curang', 'mengakui', 'pemimpin', 'hasil', 'kecurangan', 'beradab']	tetap lawan tolak pemilu curang aku pimpin hasil curang adab
1444	['minta', 'kebijaksanaan', 'mui', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'umat', 'islam', 'dituntut', 'adil', 'cc']	minta bijaksana mui tuai kyai kena hasil pemilu umat islam tuntut adil cc
1446	['kok', 'ngumumin', 'hasil', 'pemilu', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'rata', 'tidur', 'tarwih']	kok ngumumin hasil pemilu penting tengah malem bangett dahh orangorang rata tidur tarwih
1447	['hasil', 'pemilu', 'kali', 'mengetahui', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'suka', 'perdebatan', 'berdasarkan', 'teori', 'data', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lebih', 'suka', 'kekerasan', 'cont']	hasil pemilu kali tahu mayoritas duduk indonesia minim literasi suka debat dasar teori data fakta cenderung duniawi sekarang sekarang buat besok pikirinnya besok aja lebih suka keras cont

Fig 6. Stemming Result

C. LEXICON BASED LABELLING

At this stage, labeling of text data that has been previously processed will be carried out using a lexicon-based approach. At this stage, a dictionary has been prepared containing the words positive sentiment and also negative sentiment. After labeling is carried out, the data distribution for each positive, negative and neutral sentiment is as follows.

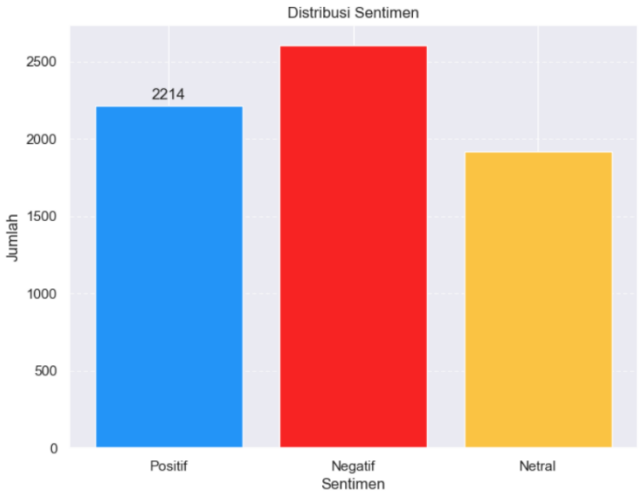


Fig 7. Sentiment Distribution

D. FEATURE EXTRACTION TF-IDF

In this process, feature extraction is carried out using the Term Frequency-Inverse Document Frequency (TF-IDF) method to convert text data into a numerical format that can be processed by a machine learning model. The results of the TF-IDF process produce 7256 features or words which have their respective weights in vector form. Figure 4.12 is an example of TF-IDF features and their weights in each document. The columns in the table represent each word in the entire sentence, while each row represents the sequence of the document or text

	bansos	curang	indonesia	mahkamah	pemilu	pilpres	prabowo	tolak
2993	0.000000	0.278981	0.169016	0.000000	0.126922	0.000000	0.000000	0.000000
2994	0.000000	0.000000	0.000000	0.168127	0.000000	0.121763	0.000000	0.000000
2995	0.000000	0.000000	0.000000	0.205216	0.000000	0.074312	0.000000	0.000000
2996	0.000000	0.000000	0.000000	0.000000	0.000000	0.124070	0.196398	0.000000
2997	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.165893	0.000000
2998	0.000000	0.000000	0.186157	0.126858	0.000000	0.091874	0.000000	0.000000
2999	0.000000	0.000000	0.000000	0.000000	0.109154	0.000000	0.227116	0.000000

Fig 8. Word in the entire sentence

E. STACKING MODELLING

At this stage, a sentiment analysis model will be built using the ensemble learning stacking method involving three algorithms as the base model, namely Naive Bayes, Random Forest, and Support Vector Machine, as well as Random Forest as a meta model. The data used in creating this model is divided in a ratio of 80:20, where 80% of the data is used for training and 20% for testing. The result is 5389 training data and 1348 testing data. The following is sample data

	text_after_stemming	sentiment
150	hiv indonesia raya tingkat azab praroro nepo baby pemenang pemilu	Negatif
151	terus lampias marah presiden kalah pilpres hasil resmi kpu rubah apa presiden wakil presiden pilih sah	Netral
152	paman mahkamah konstitusi mbatalin hasil pemilu wakanda wakanda	Negatif
153	tidak banjir pesisir utara minggu minggu ganjar duluan terjun asih bantu presiden capres pemenang pemilu tidak nilai sombong rakyat	Negatif
154	silaturahmi maksud politisasi terima hasil pemilu	Positif
155	kuat argumen mahkamah konstitusi tidak rubah hasil pilpres gandeng pilpres politikus malin kundang anak mantu cundang	Netral
156	temu tidak hak tuan rumah anies tolak hak mas gibran temu anies sosok anies diri diri sengketa hasil pilpres belum selesai	Negatif
157	kpu umum pemenang pilpres pasang prabowo gibran menang jokowi versi secc akun	Positif
158	ambil contoh pp muhammadiyah sikap hasil pemilu dewasa sabar	Positif
159	bangga juara pileg tidak terima hasil pilpres otak kerdil	Netral

Fig 9. Training Data Sample

	text_after_stemming	sentiment
150	agam hasil pemilu cermin agam masyarakat hargai	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	Negatif
152	mahkamah konstitusi pas rapat rekapitulasi manis tidak tau umum pilpres tau mahkamah konstitusi timses ngumpulin bukti tidak salah pakai advokat kurang	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	Positif
154	malam ngobrolin hasil pemilu pajak pacar gara rekonsiliasi israel	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	Positif
156	surya paloh terima hasil pemilu sahabat presiden	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	Positif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	Negatif
159	tiktok ribut hasil pemilu kali ribut hasil pemilu ribut kpop sosmed irl ngantuk kerja stngh	Netral

Fig 10. Testing Data Sample

After the data is shared, each base model will be trained using the training data and will later produce predictions on the test data. Through the predict_proba function, each model will provide a probability for whether the data is labeled positive, negative or neutral. The class or label that has the highest probability will be used as the final prediction of the model

```
# Membuat meta-feature untuk training data
nb_train_pred = nb.predict_proba(X_train_tfidf)
rf_train_pred = rf.predict_proba(X_train_tfidf)
svm_train_pred = svm.predict_proba(X_train_tfidf)
meta_features_train = np.concatenate([nb_train_pred, rf_train_pred, svm_train_pred], axis=1)

# Membuat meta-feature untuk test data
nb_test_pred_proba = nb.predict_proba(X_test_tfidf)
rf_test_pred_proba = rf.predict_proba(X_test_tfidf)
svm_test_pred_proba = svm.predict_proba(X_test_tfidf)
meta_features_test = np.concatenate([nb_test_pred_proba, rf_test_pred_proba, svm_test_pred_proba], axis=1)

# Inisialisasi dan training meta learner
meta_learner = RandomForestClassifier(n_estimators=100, random_state=42)
meta_learner.fit(meta_features_train, y_train)

# Prediksi akhir dengan meta learner pada test data
final_predictions = meta_learner.predict(meta_features_test)
```

Fig 11. Stacking Code

Figure 2 shows the prediction results and the class probability of each model against the test data. The order of classes 0, 1 and 2 in the table shows the negative, neutral and positive classes

	text	nb_class0	nb_class1	nb_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.016227	0.027230	0.956543	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.622731	0.188593	0.188675	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.348763	0.402374	0.248863	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.360907	0.252019	0.387073	Positif
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.185071	0.594748	0.220181	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.300622	0.151513	0.547865	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.015130	0.032929	0.951941	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.564754	0.209396	0.225850	Negatif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.620861	0.141082	0.238057	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.427569	0.341746	0.230685	Negatif

Fig 12. Naïve Bayes Probability Result

	text	rf_class0	rf_class1	rf_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.081357	0.262206	0.656437	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.548884	0.226526	0.224590	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.129437	0.766280	0.104283	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.102069	0.206751	0.691180	Positif
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.455056	0.416384	0.128561	Negatif
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.277810	0.125952	0.596238	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.090937	0.169992	0.739071	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.314492	0.410989	0.274519	Netral
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.411056	0.261349	0.327595	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.675032	0.170857	0.154111	Negatif

Fig 13. Random Forest Probability Result

	text	svm_class0	svm_class1	svm_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.015999	0.139643	0.844358	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.742923	0.206424	0.050653	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.134347	0.786524	0.079129	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.023798	0.508969	0.467233	Netral
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.138457	0.789167	0.072376	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.002719	0.012429	0.984851	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.000480	0.015403	0.984117	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.411272	0.226814	0.361914	Negatif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.515261	0.302810	0.181929	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.461814	0.378116	0.160070	Negatif

Fig 14. SVM Probability Result

After all base models have their own predictions, the prediction results will be combined and used as features of the meta model. So the meta model will carry out training and testing data using these new features. The following is an example of a feature that will be used by the meta model to make final predictions

	nb_class0	nb_class1	nb_class2	rf_class0	rf_class1	rf_class2	svm_class0	svm_class1	svm_class2	predict
150	0.016227	0.027230	0.956543	0.081357	0.262206	0.656437	0.015628	0.135144	0.849228	Positif
151	0.622731	0.188593	0.188675	0.548884	0.226526	0.224590	0.742093	0.207783	0.050124	Negatif
152	0.348763	0.402374	0.248863	0.129437	0.766280	0.104283	0.137193	0.784752	0.078055	Netral
153	0.360907	0.252019	0.387073	0.102069	0.206751	0.691180	0.023476	0.507070	0.469453	Positif
154	0.185071	0.594748	0.220181	0.455056	0.416384	0.128561	0.141315	0.787167	0.071518	Negatif
155	0.300622	0.151513	0.547865	0.277810	0.125952	0.596238	0.002552	0.011500	0.985948	Positif
156	0.015130	0.032929	0.951941	0.090937	0.169992	0.739071	0.000436	0.014297	0.985267	Positif
157	0.564754	0.209396	0.225850	0.314492	0.410989	0.274519	0.410736	0.224927	0.364337	Netral
158	0.620861	0.141082	0.238057	0.411056	0.261349	0.327595	0.515909	0.302254	0.181837	Positif
159	0.427569	0.341746	0.230685	0.675032	0.170857	0.154111	0.462122	0.378279	0.159599	Negatif

Fig 15. Stacking Dataset And Result

F. EVALUATION

At this stage, all models that have been trained and have produced predictions will be evaluated to see their performance using the Confusion Matrix. The evaluation matrix used includes accuracy, precision recall, and F1-score. Each matrix will be calculated using the following formula

The Confusion Matrix for the Naïve Bayes model shows that the model succeeded in predicting correctly (True Positive) 323 data with positive sentiment, 105 data with neutral sentiment, 473 data with negative sentiment

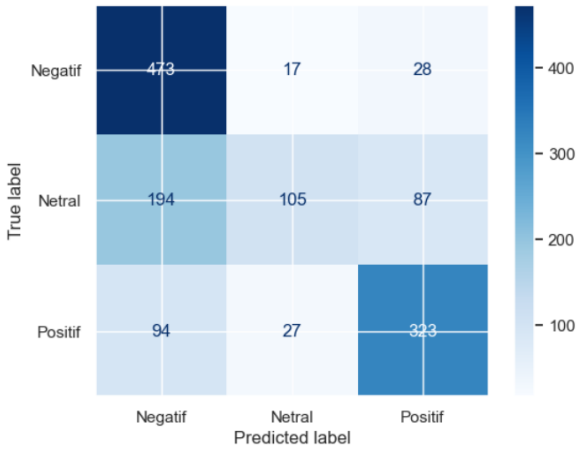


Fig 16. Naïve bayes Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the Naïve Bayes model

$$Accuracy = \frac{323 + 105 + 403}{1348} = \frac{829}{1348} = 0.6684$$

The accuracy of the Naïve Bayes model is 0.6684

$$Precission\ Positif = \frac{323}{323 + 115} = \frac{323}{438} = 0.7374$$

$$Precission\ Netral = \frac{105}{105 + 44} = \frac{105}{149} = 0{,}7047$$

$$Precission\ Negatif = \frac{473}{473 + 288} = \frac{473}{761} = 0{,}6216$$

$$Precission_{weighted} = \frac{(444 \times 0{,}7347) + (386 \times 0{,}7047) + (518 \times 0{,}6216)}{444 + 386 + 518} = 0.6835$$

The precission of the Naïve Bayes model is 0.6835

$$Recall\ Positif = \frac{323}{323 + 121} = \frac{323}{444} = 0{,}7275$$

$$Recall\ Netral = \frac{105}{105 + 281} = \frac{105}{386} = 0.272$$

$$Recall\ Negatif = \frac{473}{473+45} = \frac{473}{518} = 0.9131$$

$$Recall_{weighted} = \frac{(444 \times 0{,}7275) + (386 \times 0.272) + (518 \times 0.9131)}{444+386+518} = 0,6684$$

The recall of the Naïve Bayes model is 0.6684

$$F-1\ Score\ Positif = 2\ x\ \frac{(0{,}7374 \times 0{,}7275)}{(0{,}7374+0{,}7275)} = 0{,}7324$$

$$F-1\ Score\ Netral = 2\ x\ \frac{(0{,}7047 \times 0{,}272)}{(0{,}7047+0{,}272)} = 0{,}3925$$

$$F-1\ Score\ Negatif = 2\ x\ \frac{(0{,}6216 \times 0{,}9131)}{(0{,}6216+0{,}9131)} = 0{,}7397$$

$$F-1Score_{weighted} = \frac{(444 \times 0{,}7324) + (386 \times 0.3925) + (518 \times 0.7397)}{444+386+518} = 0,6379$$

The F-1 Score of the Naïve Bayes model is 0.6379

The Confusion Matrix for the Random Forest model shows that the model succeeded in predicting correctly (True Positive) 348 data with positive sentiment, 232 data with neutral sentiment, 428 data with negative sentiment

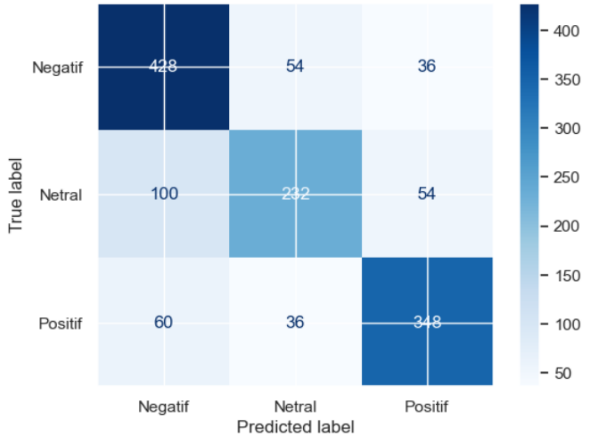


Fig 17. Random Forest Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the Random Forest model

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{348+232+428}{1348} = \frac{1008}{1348} = 0.7478$$

The accuracy of the Random Forest model is 0.7478

$$Precission\ Positif = \frac{348}{348+90} = \frac{348}{438} = 0.7945$$

$$Precission\ Netral = \frac{232}{232+90} = \frac{232}{222} = 0{,}7205$$

$$Precission\ Negatif = \frac{428}{428+160} = \frac{428}{588} = 0{,}7279$$

$$Precission_{weighted} = \frac{(444 \times 0{,}7945) + (386 \times 0{,}7205) + (518 \times 0{,}7279)}{444+386+518} = 0.7477$$

The precission of the Random Forest model is 0.7477

$$Recall\ Positif = \frac{348}{348+96} = \frac{348}{444} = 0{,}7838$$

$$Recall\ Netral = \frac{232}{232+154} = \frac{232}{386} = 0.601$$

$$Recall\ Negatif = \frac{428}{428+90} = \frac{428}{518} = 0.8263$$

$$Recall_{weighted} = \frac{(444 \times 0{,}7838) + (386 \times 0.601) + (518 \times 0.8263)}{444+386+518} = 0,7478$$

The recall of the Random Forest model is 0.7478

$$F-1\ Score\ Positif = 2\ x\ \frac{(0{,}7945 \times 0{,}7838)}{(0{,}7945+0{,}7838)} = 0{,}7891$$

$$F-1\ Score\ Netral = 2\ x\ \frac{(0{,}7205 \times 0{,}601)}{(0{,}7205+0{,}601)} = 0{,}6553$$

$$F-1\ Score\ Negatif = 2\ x\ \frac{(0{,}7279 \times 0{,}8263)}{(0{,}7279+0{,}8263)} = 0{,}774$$

$$F-1 Score_{weighted} = \frac{(444 \times 0{,}6718) + (386 \times 0.5771) + (518 \times 0.7157)}{444+386+518} = 0,745$$

The F-1 Score of the Random Forest model is 0.745

The Confusion Matrix for the SVM model shows that the model succeeded in predicting correctly (True Positive) 370 data with positive sentiment, 252 data with neutral sentiment, 426 data with negative sentiment

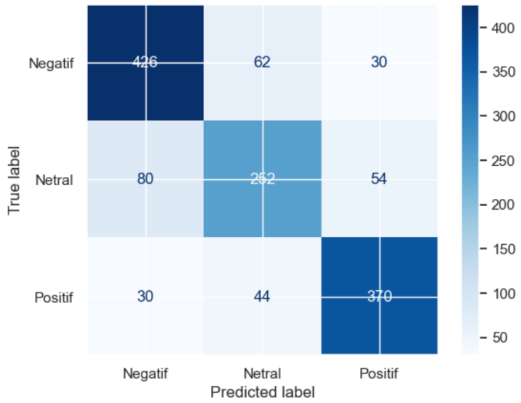


Fig 18. SVM Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the SVM model

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{370+252+426}{13512} = \frac{1048}{1351} = 0.7774$$

The accuracy of the SVM model is 0.7774

$$Precission\ Positif = \frac{370}{370+84} = \frac{370}{454} = 0.815$$

$$Precission\ Netral = \frac{252}{252+106} = \frac{252}{358} = 0{,}7039$$

$$Precission\ Negatif = \frac{426}{426+110} = \frac{426}{536} = 0{,}7984$$

$$Precission_{weighted} = \frac{(444 \times 0{,}815)\ + (386 \times 0{,}7039) + (518 \times 0{,}7984)}{444+386+518} = 0.7754$$

The precission of the naïve Bayes model is 0.7754

$$Recall\ Positif = \frac{370}{370+74} = \frac{370}{444} = 0{,}8333$$

$$Recall\ Netral = \frac{252}{252+134} = \frac{252}{386} = 0.6528$$

$$Recall\ Negatif = \frac{426}{426+92} = \frac{426}{518} = 0.8224$$

$$Recall_{weighted} = \frac{(444 \times 0{,}8333) + (386 \times 0.6528) + (518 \times 0.8224)}{444+386+518} = 0,7774$$

The recall of the SVM model is 0.7774

$$F-1\ Score\ Positif = 2\ x\ \frac{(0{,}815 \times 0{,}8333)}{(0{,}815+0{,}8333)} = 0{,}824$$

$$F-1\ Score\ Netral = 2\ x\ \frac{(0{,}7039 \times 0{,}6528)}{(0{,}7039+0{,}6528)} = 0{,}6774$$

$$F-1\ Score\ Negatif = 2\ x\ \frac{(0{,}7984 \times 0{,}8224)}{(0{,}7984+0{,}8224)} = 0{,}8084$$

$$F-1Score_{weighted} = \frac{(444 \times 0{,}824) + (386 \times 0.6774) + (518 \times 0.8084)}{444+386+518} = 0,8084$$

The F-1 Score of the SVM model is 0.8084

The Confusion Matrix for the stacking model shows that the model succeeded in predicting correctly (True Positive) 396 data with positive sentiment, 256 data with neutral sentiment, 447 data with negative sentiment.

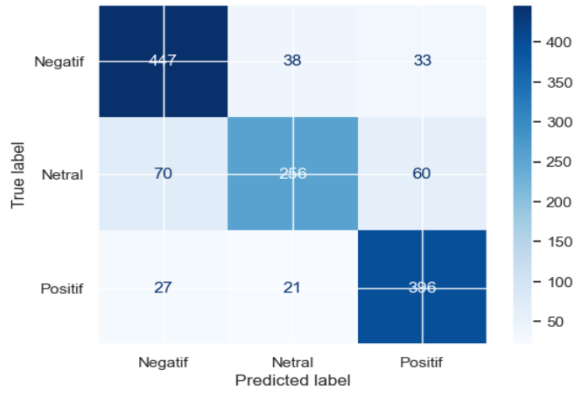


Fig 19. Stacking Model's Confussion Matrix

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{396+256+447}{13512} = \frac{1099}{1351} = 0.8153$$

Below are calculations to find the accuracy, precision, recall, and F1-score of the Stacking model

$$Precission\ Positif = \frac{396}{396+93} = \frac{396}{489} = 0.8089$$

$$Precission\ Netral = \frac{256}{256+59} = \frac{256}{315} = 0.8127$$

$$Precission\ Negatif = \frac{447}{447+97} = \frac{447}{544} = 0{,}8127$$

$$Precission_{weighted} = \frac{(444 \times 0{,}8089) + (386 \times 0{,}8127) + (518 \times 0{,}8127)}{444+386+518} = 0.8152$$

The accuracy of the Stacking model is 0.8152

$$Recall\ Positif = \frac{396}{396+48} = \frac{396}{444} = 0{,}8919$$

$$Recall\ Netral = \frac{256}{256+130} = \frac{256}{386} = 0.6632$$

$$Recall\ Negatif = \frac{447}{447+71} = \frac{447}{518} = 0.8629$$

$$Recall_{weighted} = \frac{(444 \times 0{,}8919) + (386 \times 0.6632) + (518 \times 0.8629)}{444+386+518} = 0,8153$$

The recall of the Stacking model is 0.8153

$$F-1\ Score\ Positif\ =\ 2\ x\ \frac{(0{,}8098 \times 0{,}8919)}{(0{,}8098 + 0{,}8919)} = 0{,}8489$$

$$F-1\ Score\ Netral\ =\ 2\ x\ \frac{(0{,}8127 \times 0{,}6632)}{(0{,}8127 + 0{,}6632)} = 0{,}7304$$

$$F-1\ Score\ Negatif\ =\ 2\ x\ \frac{(0{,}8217 \times 0{,}8629)}{(0{,}8217 + 0{,}8629)} = 0{,}8418$$

$$F-1 Score_{weighted} = \frac{(444 \times 0{,}6718) + (386 \times 0.5771) + (518 \times 0.7157)}{444 + 386 + 518} = 0,8122$$

The F-1 Score of the Stacking model is 0.8122

The Result of all model can be seen in the table below

Table 3. Results Of Model

Model	Accuracy	Precission	Recall	F-1 Score
Naïve Bayes	0.6684	0.6835	0,6684	0,6379
Support Vector Machine	0.7774	0.7754	0,7774	0,776
Random Forest	0.7478	0.7477	0,7478	0,745
Ensemble Stacking (RF)	0.8153	0.8152	0,8153	0,8122

4. CONCLUSION

As a result of this experiment, an ensemble learning stacking model was formed with several different base models, namely the SVM, Random Forest and Naïve Bayes algorithms. Each model carries out training and predictions on sentiment analysis data. The results, starting from the lowest, are the Naïve Bayes algorithm with an accuracy of 66.84%, followed by Random Forest with an accuracy of 74.78%, and the highest is SVM with an accuracy of 77.74%. The results of the three base models are compiled and used as input for a meta model that uses the Random Forest algorithm. The results show that the stacking ensemble method applied produces better accuracy than a single classifier, namely 81.53%. Thus, it can be concluded that ensemble learning stacking with the SVM, Random Forest and Naïve Bayes base models as well as meta models using Random Forest can increase the accuracy of sentiment analysis models on unstructured text data

REFERENCES

[1] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, 2023, doi: 10.1016/j.jksuci.2023.01.014.

[2] F. Matloob *et al.*, "Software defect prediction using ensemble learning: A systematic literature review," *IEEE Access*, vol. 9, pp. 98754–98771, 2021, doi: 10.1109/ACCESS.2021.3095559.

[3] Y. Görmez, Y. E. Işık, M. Temiz, and Z. Aydın, "FBSEM: A Novel Feature-Based Stacked Ensemble Method for Sentiment Analysis," *Int. J. Inf. Technol. Comput. Sci.*, vol. 12, no. 6, pp. 11–22, 2020, doi: 10.5815/ijitcs.2020.06.02.

[4] P. Thiengburanathum and P. Charoenkwan, "SETAR: Stacking Ensemble Learning for Thai Sentiment Analysis Using RoBERTa and Hybrid Feature Representation," *IEEE Access*, vol. 11, no. August, pp. 92822–92837, 2023, doi: 10.1109/ACCESS.2023.3308951.

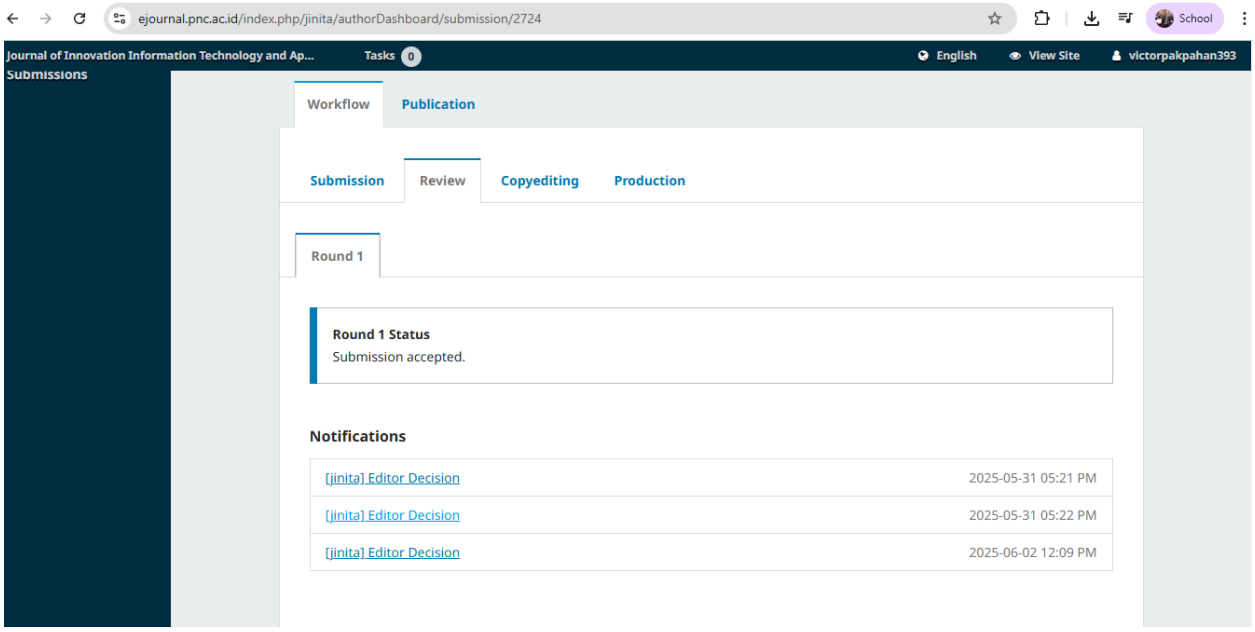
[5] J. Gu, S. Liu, Z. Zhou, S. R. Chalov, and Q. Zhuang, "A Stacking Ensemble Learning Model for Monthly Rainfall Prediction in the Taihu Basin, China," *Water (Switzerland)*, vol. 14, no. 3, 2022, doi: 10.3390/w14030492.

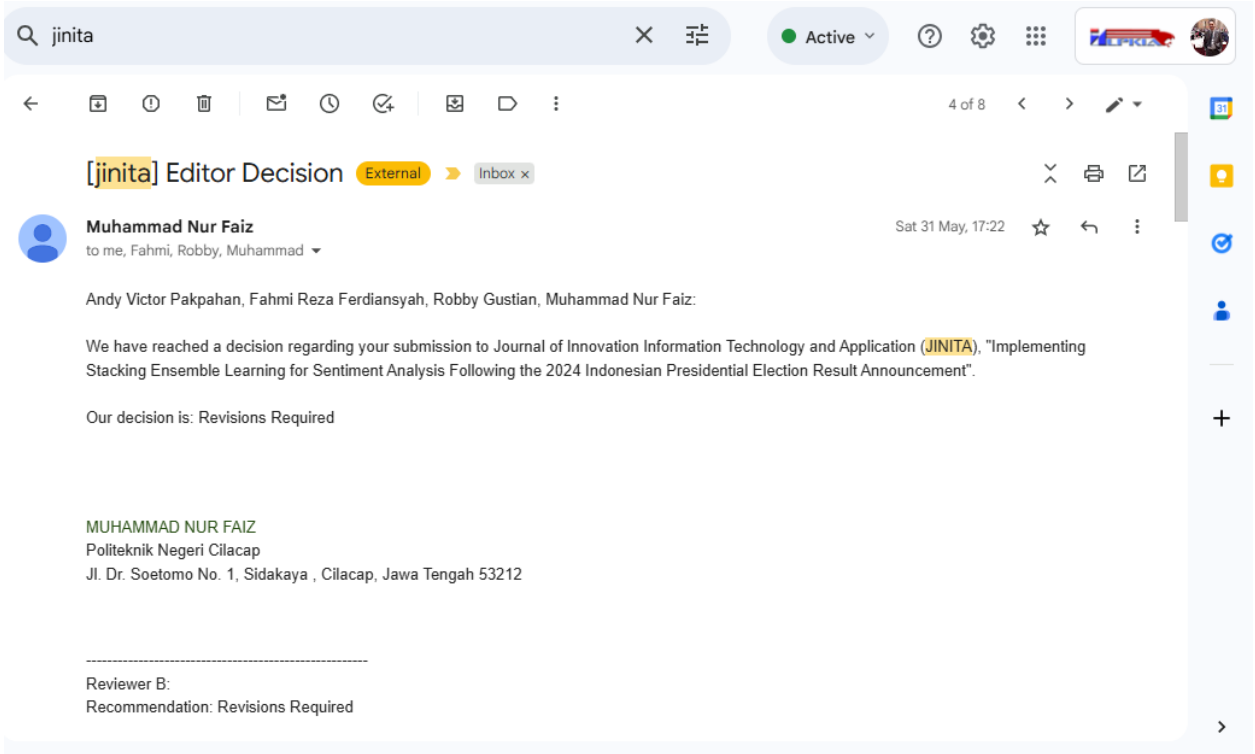
[6] S. A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, "Stacking Strong Ensembles of Classifiers," in *IFIP Advances in Information and Communication Technology*, 2019. doi: 10.1007/978-3-030-19823-7_46.

[7] J. Dou *et al.*, "Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan," *Landslides*, vol. 17, no.

3, 2020, doi: 10.1007/s10346-019-01286-5.
[8] I. D. Mienye and Y. Sun, “A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects,” *IEEE Access*, vol. 10, no. September, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
[9] M. B. Alfarazi, M. ’Ariful Furqon, and H. Soepandi, “Sentiment Analysis of User Reviews for the Sister For Student Application Using Gaussian Naive Bayes and N-Gram,” *J. Innov. Inf. Technol. Appl.*, pp. 101–108, 2024.
[10] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artif. Intell. Rev.*, vol. 55, no. 7, 2022, doi: 10.1007/s10462-022-10144-1.
[11] A. M. Iddrisu, S. Mensah, F. Boafo, G. R. Yeluripati, and P. Kudjo, “A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 2, 2023, doi: 10.1016/j.jjimei.2023.100180.
[12] R. Michaela Denise Gonzales and C. A. Hargreaves, “How can we use artificial intelligence for stock recommendation and risk management? A proposed decision support system,” *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, 2022, doi: 10.1016/j.jjimei.2022.100130.
[13] A. Tripathi, T. Goswami, S. K. Trivedi, and R. D. Sharma, “A multi class random forest (MCRF) model for classification of small plant peptides,” *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100029, 2021, doi: 10.1016/j.jjimei.2021.100029.
[14] M. Y. Aldean, P. Paradise, and N. A. Setya Nugraha, “Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 di Twitter Menggunakan Metode Random Forest Classifier (Studi Kasus: Vaksin Sinovac),” *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 4, no. 2, pp. 64–72, 2022, doi: 10.20895/inista.v4i2.575.
[15] A. Syafrianto, “Perbandingan Algoritma Naïve Bayes dan Decision Tree Pada Sentimen Analisis,” *Indones. J. Comput. Sci. Res.*, vol. 1, no. 2, pp. 1–15, 2022, doi: 10.59095/ijcsr.v1i2.11.
[16] R. Nuraeni, A. Sudiarjo, and R. Rizal, “Perbandingan Algoritma Naïve Bayes Classifier dan Algoritma Decision Tree untuk Analisa Sistem Klasifikasi Judul Skripsi,” *Innov. Res. Informatics*, vol. 3, no. 1, pp. 26–31, 2021, doi: 10.37058/innovatics.v3i1.2976.
[17] S. H. Ramadhani and M. I. Wahyudin, “Analisis Sentimen Terhadap Vaksinasi Astra Zeneca pada Twitter Menggunakan Metode Naïve Bayes dan K-NN,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 4, 2022, doi: 10.35870/jtik.v6i4.530.
[18] M. Asgari, W. Yang, and M. Farnaghi, “Spatiotemporal data partitioning for distributed random forest algorithm: Air quality prediction using imbalanced big spatiotemporal data on spark distributed framework,” *Environ. Technol. Innov.*, vol. 27, p. 102776, 2022, doi: 10.1016/j.eti.2022.102776.
[19] G. Meena, K. K. Mohbey, and S. Kumar, “Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 1, p. 100174, 2023, doi: 10.1016/j.jjimei.2023.100174.
[20] K. L. Tan, C. P. Lee, and K. M. Lim, “A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research,” *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074550.

4. JINITA Editor Decision – Revisions Required





Andy Victor <abang@lpkia.ac.id>

[jinita] Editor Decision

2 messages

Muhammad Nur Faiz <ejournal.pnc@gmail.com> 31 May 2025 at 17:21
To: Andy Victor Pakpahan <abang@lpkia.ac.id>, Fahmi Reza Ferdiansyah <fahmirezaf@lpkia.ac.id>, Robby Gustian <230434004@fellow.lpkia.ac.id>, Muhammad Nur Faiz <faiz@pnc.ac.id>

Andy Victor Pakpahan, Fahmi Reza Ferdiansyah, Robby Gustian, Muhammad Nur Faiz:

We have reached a decision regarding your submission to Journal of Innovation Information Technology and Application (JINITA), "Implementing Stacking Ensemble Learning for Sentiment Analysis Following the 2024 Indonesian Presidential Election Result Announcement".

Our decision is: Revisions Required

MUHAMMAD NUR FAIZ
Politeknik Negeri Cilacap
Jl. Dr. Soetomo No. 1, Sidakaya , Cilacap, Jawa Tengah 53212

Reviewer B:
Recommendation: Revisions Required

Does the title match the substance of the article?

> Yes, the title of the article matches the substance of the article.
> The article's title accurately represents its content, which centers on applying stacking ensemble learning for sentiment analysis of public opinion on the 2024 Indonesian presidential election results. The study employs Naïve Bayes, SVM, and Random Forest as base models, with Random Forest as the meta-model, using 6,737 tweets collected via web scraping. It details each research stage—data preprocessing, lexicon-based sentiment labeling, TF-IDF feature extraction, and performance evaluation. The stacking model achieved the highest accuracy of 81.53%, confirming that the title effectively reflects the study's scope and findings.

Does the introduction provide sufficient background and clearly justify the study?

> Yes, the introduction provides sufficient background and clearly justifies the study.
> The introduction effectively combines theoretical background, problem identification, justification of methods, and a clear research aim. Therefore, it is well-structured and sufficiently justifies the purpose and direction of the study.

Does the paper demonstrate an adequate understanding of the relevant literature in the field and cite an appropriate range of literature sources?

> Yes, the paper demonstrates an adequate understanding of the relevant literature and cites an appropriate range of sources.
> The paper demonstrates adequate understanding of the relevant literature by referencing a diverse and up-to-date range of sources related to ensemble learning, stacking methods, and sentiment analysis. It cites studies that justify the selection of algorithms and highlight their comparative performance. The literature used is relevant, recent, and supports both the theoretical framework and the research methodology.

Are the methods adequately described?

Yes, the methods are adequately described in the paper.
The paper provides a clear and detailed description of the methods used, covering each step of the research process. It explains the data collection via web scraping, preprocessing techniques (including text cleaning, tokenization, stopword removal, and stemming), sentiment labeling using a lexicon-based approach, and feature extraction using TF-IDF. The stacking ensemble modeling is thoroughly outlined, including base and meta-model selection, data splitting, and prediction procedures. Additionally, the evaluation metrics (accuracy, precision, recall, and F1-score) are clearly defined. Overall, the methodology is transparent, systematic, and replicable.

Are the results clearly presented?

Yes, the results are clearly presented in the paper.
The results are presented in a structured and comprehensible manner, using both textual explanations and visual aids such as tables and figures. Performance metrics for each model—Naïve Bayes, SVM, Random Forest, and the Stacking ensemble—are clearly reported, including accuracy, precision, recall, and F1-score. Confusion matrices and probability outputs are also included to illustrate model predictions. A comparison table summarizes all model performances, making it easy for readers to interpret and compare results. Overall, the presentation of results is clear, complete, and supports the study's conclusions.

Are the conclusions supported by the results?

Yes, the conclusions in the paper are supported by the results.
The study concludes that the stacking ensemble method—combining Naïve Bayes, SVM, and Random Forest as base models with Random Forest as the meta-model—achieves the highest accuracy (81.53%) compared to individual models. This conclusion is supported by thorough evaluation metrics and a solid methodology, including data preprocessing, lexicon-based labeling, TF-IDF feature extraction, and confusion matrix analysis. The results confirm that stacking improves sentiment analysis performance on unstructured Twitter data.

Does the paper identify clearly any implications for research, practice and/or society?

The paper does not explicitly state the implications for research, practice, or society in a dedicated section or clear language.
While the paper effectively demonstrates technical success, it would benefit from a more explicit discussion on the broader impact and real-world relevance of its findings.

Recommendations

Accept with minor Revision

Comments and Suggestions for Authors:

Suggestions for Improvement (Point by Point)
1. Language and Grammar
a. Correct spelling and grammatical errors, such as "precission" → should be precision, "repliess" → replies, "Ouput" → Output, etc.
b. Several sentences are grammatically incorrect. It's recommended to use a grammar checker or have the paper reviewed by a native
English speaker or professional editor.

2. Scientific Structure.
a. Include a dedicated "Limitations" or "Discussion" section to address aspects such as: The reliance on lexicon-based labeling, Possible
overfitting risks in the stacking model, Potential class imbalance in the sentiment data.
b. The IEEE-style citation template is incomplete (e.g., "Article Title" is left as a placeholder). This needs to be fixed.

3. Reproducibility and Transparency
a. The paper does not mention whether the dataset or code will be made publicly available. It is advisable to: Provide a GitHub
repository or data-sharing link, Mention if the code/scripts can be accessed for replication purposes.

4. Deeper Sentiment Analysis
a. Add insights on sentiment distribution based on time range or keyword categories. This would enhance the contextual
understanding of the political discourse.
b. Include examples of tweets and classification outcomes to illustrate how the model performs in real-world cases.

5. Figures and Visual Quality
a. Some figures (e.g., stemming and tokenization results) lack clarity or informative captions. Improve image quality and provide better
explanations for each visual element.

Reviewer D:
Recommendation: Revisions Required

Does the title match the substance of the article?

Yes, it does.

Does the introduction provide sufficient background and clearly justify the study?

Yes, it does.

Does the paper demonstrate an adequate understanding of the relevant literature in the field and cite an appropriate range of literature sources?

Yes, it does.

Are the methods adequately described?

Yes, they are.

Are the results clearly presented?

Yes, they are.

Are the conclusions supported by the results?

Yes, they are.

Does the paper identify clearly any implications for research, practice and/or society?

Yes, it does.

Recommendations

Accept with minor Revision

Comments and Suggestions for Authors:

The research has shown a good contribution. However, it needs to be revised in accordance with my suggestions for improvement so that the research can have a better impact.

Journal of Innovation Information Technology and Application (JINITA)
https://ejournal.pnc.ac.id/index.php/jinita

Muhammad Nur Faiz <ejournal.pnc@gmail.com> 31 May 2025 at 17:22
To: Andy Victor Pakpahan <abang@lpkia.ac.id>, Fahmi Reza Ferdiansyah <fahmirezaf@lpkia.ac.id>, Robby Gustian <230434004@fellow.lpkia.ac.id>, Muhammad Nur Faiz <faiz@pnc.ac.id>

[Quoted text hidden]

Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Implementing Stacking Ensemble Learning for Sentiment Analysis Following the 2024 Indonesian Presidential Election Result Announcement

Commented [HS1]: In this section, the sentence length should not be more than 12 words.

ARTICLE INFO

Article history:
Received 24 January 2020
Revised 30 April 2020
Accepted 2 December 2020
Available online xxx

Keywords:
Ensemble Learning,
Stacking,
Sentiment Analysis,
Data Mining,
Machine Learning

IEEE style in citing this article:
Jinita and J. Jinita, "Article Title," *Journal of Innovation Information Technology and Application*, vol. 4, no. 1, pp. 1-10, 2022. [Fill citation heading]

ABSTRACT

Sentiment analysis is a technique for processing text with the aim of identifying opinions and emotions within a sentence. In its approach, machine learning has become a commonly used method. Several algorithms such as Naive Bayes, Support Vector Machine (SVM), and Random Forest are often used in this analysis. However, creating a model that has optimal accuracy is still a challenge, especially for sentiment analysis of unstructured text data. The aim of this research is to try to improve the accuracy of the sentiment analysis model by using the ensemble learning stacking method approach, namely a method for combining several base models to produce better model performance. The algorithms that will be the base models in this stacking are Random Forest, Naive Bayes and SVM, while the model that makes the final prediction or what is called a meta model will use Random Forest. This sentiment analysis was carried out in a specific context, namely public opinion after the announcement of the results of the 2024 Indonesian presidential election. The dataset analyzed using reviews or public opinion regarding the results of the 2024 Indonesian presidential election was 6737 tweet data collected via the X platform using web scraping. As a result, the Naïve Bayes method got an accuracy of 66.84%, SVM got an accuracy of 77.74%, Random Forest got an accuracy of 74.78% and stacking got an accuracy of 81.53%.

Commented [HS2]: In this sentence, the year should be used at the end of the sentence, for example "... in 2024."

Commented [HS3]: In this sentence, the year should be used at the end of the sentence, for example "... in 2024."

Commented [HS4]: In this sentence, it is necessary to highlight the contributions made, not only the evaluation metrics used and their percentages, but also the impact of ensemble learning, which needs to be well explained.

1. INTRODUCTION

The Ensemble Learning is a method in machine learning that combines several models to create a new model that is stronger than and has superior performance compared to when the algorithms are used individually [1], [2]. There are several ensemble learning techniques such as bagging, stacking, averaging and boosting, each technique is distinguished by how the model is trained and combined [1].

Stacking is an ensemble learning technique that works by combining the results of several different base-models. Each base-model will learn and have its own prediction results, after that a final model will be created which will combine the prediction results of all the base-models which is called a meta-model [3], [4]. The Stacking technique is based on the idea that each basic model has its own advantages and disadvantages [5]. By combining predictions from different base-models, the resulting meta-model can learn and balance these advantages and disadvantages appropriately, so that the overall performance of the stacking model can exceed the performance of any individual model and makes it a fairly good technique for improving predictive power of the classifier [6], [7]. This is the advantage of the stacking technique compared to other ensemble learning techniques and makes stacking a suitable technique for creating models for processing quite complex data such as sentiment analysis [8]

Sentiment analysis is the process of understanding, extracting and processing textual data automatically to obtain information on opinions, feelings and emotions contained in a sentence [9]. Sentiment analysis aims to understand a person's level of satisfaction and dissatisfaction with a service or product, as well as understanding public perceptions regarding a person's agreement and disagreement with a particular topic [10]

*) Corresponding Author : abang@lpkia.ac.id

Sentiment analysis is generally made using classification algorithm models such as Support Vector Machine (SVM), Decission Tree, K-Nearest-Neighbor (KNN), Naïve Bayes, Random Forest, etc [11], [12]. Several classification algorithms have been used in several previous studies regarding sentiment analysis carried out on opinions taken from social media X or Twitter in Indonesian and each algorithm has different accuracy [11]. Compared the SVM algorithm with other algorithms such as Naïve Bayes, Decission Tree and KNN in sentiment analysis with different cases or topics, the results is that SVM accuracy is better when compared to other algorithms. Even though Naïve Bayes is not superior in accuracy to the SVM algorithm, if we refer to research conducted Naïve Bayes still has better accuracy results when compared to the Decision Tree and KNN algorithms [13]. Then, if we refer to research which compared the Random Forest algorithm with other algorithms such as Naïve Bayes, KNN, Decission Tree and Logistic Regression, it can be seen that Random Forest produces better accuracy than other algorithms including SVM [14]. Other research that shows that Random Forest is superior to SVM is research From these studies, it can be seen that the Random Forest, SVM and Naïve Bayes algorithms are some of the algorithms with the best accuracy in terms of sentiment analysis.

Even so, sentiment analysis is not an easy task to do. The complexity of language and variations in human expressions in various sentences make sentiment analysis a challenge [15], [16]. Building a model that can produce accuracy and good performance is also a challenge in sentiment analysis [17] especially sentiment analysis of unstructured text, for example data taken from social media such as X or Twitter has its own challenges because the language used is usually not appropriate. standard words, involving abbreviations, as well as words that are not in the dictionary, thus affecting accuracy[18], [19]. So the accuracy of the sentiment analysis model can still be improved with the help of other methods, for example by using the ensemble stacking method.

Based on the description above, a sentiment analysis model will be built using the ensemble learning stacking method, with the aim of increasing the accuracy of the model in sentiment analysis on unstructured text, which in this case is data collected via the social media platform [10], [20]. This sentiment is the public's opinion regarding the results of the 2024 Indonesian Presidential Election. We propose to use Naive Bayes, Random Forest and Support Vector Machine as base models and Random Forest as a meta model because these models are suitable for sentiment analysis and have been widely used by previous researchers. Beside from that, these models also have different characteristics.

2. METHOD

The research flow presented in this experiment outlines the structured steps taken to achieve the objectives of the study. It begins with the identification of the problem, which serves as the foundation for formulating the research questions and determining the appropriate methodology. This initial phase is crucial to ensure that the research direction is clear and aligned with the intended goals.

Following problem identification, the flow continues through stages such as data collection, analysis, and interpretation. Each step is interconnected, allowing the process to build logically upon the previous one. This structured approach not only helps in maintaining the consistency of the study but also enhances the reliability and validity of the results obtained.

Figure 1 provides a visual representation of this research flow. It serves as a guide to understand how the experiment was conducted from start to finish. By presenting the process in a flowchart format, it becomes easier to grasp the overall methodology and appreciate the systematic effort involved in reaching the research conclusions.

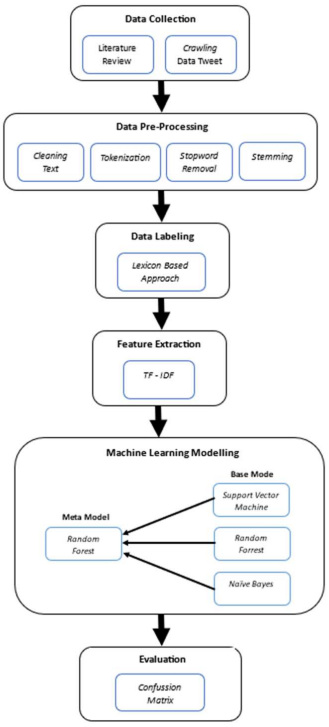


Fig 1. Flow of Research

A. DATA COLLECTION

Based on the description above, a sentiment analysis model will be built using the ensemble learning stacking method, with the aim of increasing

B. DATA PRE-PROCESSING

This process includes a series of steps to prepare the data before creating a sentiment analysis model. Stages that will be carried out in the process pre-processing data is as follows:

1) CLEANING TEXT

At this stage, text data will be cleaned has been collected from scrapping results so that the text can be made easier processed by the next stage. Data cleaning includes several processes such as deleting numbers and symbols, changing text to lowercase and also normalize the text or change each word in a sentence becomes standard or normal form for omission non-standard words, abbreviations, slang words, typo words etc. Text normalization This is done by referring to the dictionary provided which contains non-standard words and actual standard words.

2) TOKENIZATION

At this stage, every text data has been cleaned will be convert into small parts of each word in sentences are called tokens. For example, sentence “Indonesia Lebih Maju” will be converted to ["Indonesia", "lebih", "maju"]

3) STOPWORD REMOVAL

At this stage, any common words are not make significant contributions to the meaning of the text will be removed. The stopword dictionary will be taken from a library that has provided a list the stop words are

Sastrawi. Some examples of words included in the stopword and will be deleted are like “yang”, “dan”, “di”, “adalah”.

4) STEMMING

At this stage, every word in the text will be changed be the basic word. Words with the same ending or words those with affixes will be changed to the basic form.

C. DATA LABELING

The method that will be used for data labeling is Lexicon Based. Lexicon based Approach can be used to create labeled training datasets for sentiment analysis machine learning algorithms that require labels at the start his training [23]. The idea behind the lexicon based approach is that the meaning of a text is greatly influenced by the polarity of the words and phrases in inside. This includes words such as adjectives, adverbs, nouns, verbs, as well as phrases and sentences that contain them [24]. This approach makes use of a dictionary or list of words with predefined sentiment labels. Any data will be carried out Check the total score of positive words and negative words. If the word score positive exceeds negative scores, then the label is positive, and vice versa then the label is negative. However, if the score is the same or 0, then the label is will be neutral.

D. FEATURE EXTRACTION

In this process, feature extraction will be carried out using the Term Frequency-Inverse Document Frequency (TF-IDF). At this stage, every tweet will be represented as a numerical feature vector, where each component The vector will represent the weight of each word in the existing word dictionary. This weight calculated based on the frequency of occurrence of words in tweets (TF) and inverse proportional to the occurrence of the word in the entire collection of tweets (IDF). This feature extraction process aims to change the tweet text into a numerical representation that can be used by the model to perform analysis further sentiment. The formulas of TF-IDF are as follows:

$$TF(t,d) = \frac{number\ of\ occurences\ of\ word\ in\ document}{total\ number\ if\ words\ in\ document} \quad (1)$$

$$IDF(t,D) = \log\left(\frac{N}{df(t,D)}\right) \quad (2)$$

$$TF - IDF(t,d) = TF(t,d)x\ IDF(t,D) \quad (3)$$

- N : Total number of documents in the collection
- df(t,D): Number of documents in the collection containing term t
- TF(t,d): Term Frequency of term t in document d
- IDF(t,D): Inverse Document Frequency of term t in all documents D
- W(t,d): Weight of term t in a document

E. STACKING MODELLING

At this stage a model will be built for sentiment analysis. Before that, The data will first be split into 2 parts, namely training data and testing data with a percentage of 80% training data and 20% test data. Then, it will be done training on each base model, namely naïve Bayes, support vector machine and random forest use data that has been split. Every base the model will generate predictions based on these features. Then as in the ensemble learning stacking technique stage, the output from the base model will be used as a feature for the meta model model, which is in charge combining predictions from the base model to produce a prediction the new one. The following is an illustration of the stacking model that will be combined.

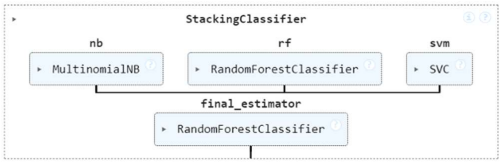


Fig 2. Stacking Model Illustration

Commented [HS5]: This sentence should mention the formula used.

Commented [HS6]: What does this part mean? If explaining the formulas, add the necessary sentences first.

Commented [HS7]: In this section, it is necessary to mention Figure 2 and Table 1 as references. In addition, it is necessary to explain Table 1 also related to the algorithm used, at least 4-5 sentences containing the main sentence and explanatory sentences.

DOI: doi.org/10.35970/jinita.vxixx.xx

Here is the algorithm for the stacking model that will be created

Table 1. Algortihm Stacking

Algorithm 1. Stacking

Input: X_train, y_train, x_test, base_models, meta_model

Ouput: prediction meta model

```
START
base_model_outputs_train = []
FOR model in base_models THEN
    model.fit(X_train, y_train)
    probas_train = model.predict_proba(X_train)
    base_model_outputs_train.append(probas_train)
END FOR
meta_features_train = np.hstack(base_model_outputs_train)
meta_model.fit(meta_features_train, y_train)
base_model_outputs_test = []
FOR model in base_models THEN
    probas_test = model.predict_proba(X_test)
    base_model_outputs_test.append(probas_test)
END FOR
meta_features_test = np.hstack(base_model_outputs_test)
final_predictions = meta_model.predict(meta_features_test)
END
```

F. EVALUATION

After the model building process is complete, the next step is evaluate model performance using confusion matrix. Evaluation is carried out against each base model and meta model itself, so you can see the comparison of classification results between models. Because this sentiment analysis has 3 labels, namely positive, negative and neutral (multi class), the calculation of each metric will use the concept of a weighted average where the formula is as follows

Commented [HS8]: This section should be bolded.

Commented [HS9]: In this section, it is necessary to mention the formulas used as references for the explanatory sentences.

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} \quad (4)$$

$$Precission_{weighted} = \frac{\sum_{i=1}^{N}(Precission_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (5)$$

$$Recall_{weighted} = \frac{\sum_{i=1}^{N}(Recall_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (6)$$

$$F-1Score_{weighted} = \frac{\sum_{i=1}^{N}(F-1score_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (7)$$

These evaluation metrics provide a comprehensive view of the model's ability to correctly classify sentiments across all classes. By using weighted averages, the metrics take into account the proportion of each class, ensuring that imbalanced class distributions do not bias the results. This is particularly important in multiclass classification problems where some classes may dominate. The use of these formulas allows for a fair comparison of performance between base models and the meta model.

3. RESULTS AND DISCUSSION

A. WEB SCRAPPING RESULT

The total data collected was 8094 tweets with details of each keyword are as follows.

Table 2. Data Collected by keyword.

Keyword	Query Search	Result
Hasil pemilu	Hasil pemilu lang:id until:2024-04-30 since:2024-03-20 -filter:links -filter:repliess	3482 tweet
Hasil pemilu presiden	Hasil pemilu presiden lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	156 tweet
Hasil pilpres	Hasil pilpres lang:id until:2024-04-30 sinc e:2024-03-20 -filter:links -filter:repliess	1983 tweet
Pemenang pemilu	Pemenang pemilu lang:id until:2024-04-3 0 since:2024-03-20 -filter:links -filter:rep liess	1000 tweet
Pemenang pilpres	Pemenang pilpres lang:id until:2024-04-3 0 since:2024-03-20 -filter:links -filter:rep liess	808 tweet
Pemenang presiden	Pemenang presiden lang:id until:2024-04 30 since:2024-03-20 -filter:links -filter:re pliess	274 tweet
Pengumuman pemilu	Pengumuman pemilu lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	221 tweet
Pengumuman pilpres	Pengumuman pilpres lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	170 tweet

B. DATA PRE-PROCESSING

The pre-processing stage is the first step in preparing the dataset by carrying out several stages, namely cleaning text, tokenization, stopword removal and stemming as well as deleting duplicate data. Data cleaning of scrapped tweet text includes several processes such as deleting mentions, deleting hashtags, deleting retweets, deleting URLs, deleting non-alphanumeric characters, deleting double spaces and transform the text into lowercase. Then, text normalization will be carried out to change words such as abbreviations, non-standard words, and slang words into normal and formal words. Finally, to avoid data duplication, tweet data that has the same or duplicate sentences will be deleted. So the final total of tweet data that will be used in the next stage until the end is 6737 tweets. Here are the results

	lowered_text	normalized_text
1440	sudah tdk kaget dgn hasil pilpres yg di umumkan kpu malam ini sdh dr awal sejak mk dan kpu meloloskan gibram menjadi calon wakil prabowo menjadi salala satu kandidat capres pemilu bak sinetron yg kita sdh tau di mna ujung ceritanya mari kita tunggu sinetron episode berikut nya	sudah tidak kaget dengan hasil pilpres yang di umumkan kpu malam ini sudah dari awal sejak mk dan kpu meloloskan gibran menjadi calon wakil prabowo menjadi salah satu kandidat capres pemilu bak sinetron yang kita sudah tau di mana ujung ceritanya mari kita tunggu sinetron episode berikut nya
1441	pemilu hasil bansos 465 t	pemilu hasil bansos 465 t
1442	pemilu kok dianggap perang kacau ente ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok nte tidak cepat cuci muka sono biar siuman	pemilu kok dianggap perang kacau anda ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok anda tidak cepat cuci muka sono biar siuman
1443	tetap lawan amp tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yg tak beradab	tetap lawan sampai tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yang tidak beradab
1444	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc
1446	kok ngumumin hasil pemilu yg amat penting tengah malem bangett dahh orangorang juga rata2 udh pada tidur abis tarwih	kok ngumumin hasil pemilu yang amat penting tengah malem bangett dahh orangorang juga rata2 sudah pada tidur abis tarwih
1447	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yg berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yang berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont

Fig 3. Cleaning Text Result

Next, the tweet text that has been cleaned will be broken down into pieces of words in sentences called tokens. Following are some results from the tokenization process

Commented [HS10]: In the table format section, it needs to be adjusted to the provisions in the JINITA journal.

Commented [HS11]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

Commented [HS12]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

DOI: doi.org/10.35970/jinita.vxixx.xx

	normalized_text	tokenized_text
1440	sudah tidak kaget dengan hasil pilpres yang di umumkan kpu malam ini sudah dari awal sejak mk dan kpu meloloskan gibran menjadi calon wakil prabowo menjadi salah satu kandidat capres pemilu bak sinetron yang kita sudah tau di mana ujung ceritanya mari kita tunggu sinetron episode berikut nya	['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini', 'sudah', 'dari', 'awal', 'sejak', 'mk', 'dan', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'yang', 'kita', 'sudah', 'tau', 'di', 'mana', 'ujung', 'ceritanya', 'mari', 'kita', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']
1441	pemilu hasil bansos 465 t	['pemilu', 'hasil', 'bansos', '465', 't']
1442	pemilu kok dianggap perang kacau anda ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok anda tidak cepat cuci muka sono biar siuman	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'anda', 'ini', 'ketum', 'nasdem', 'saja', 'sebagai', 'partai', 'penyokong', 'utama', 'paslon', 'no', '1', 'sudah', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'anda', 'tidak', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']
1443	tetap lawan sampai tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yang tidak beradab	['tetap', 'lawan', 'sampai', 'tolak', 'pemilu', 'curang', 'tidak', 'mengakui', 'pemimpin', 'dari', 'hasil', 'kecurangan', 'yang', 'tidak', 'beradab']
1444	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc	['saya', 'minta', 'kebijaksanaan', 'mui', 'yang', 'saat', 'ini', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', '2024', 'sebagai', 'umat', 'islam', 'kita', 'dituntut', 'untuk', 'adil', 'cc']
1446	kok ngumumin hasil pemilu yang amat penting tengah malem bangett dahh orangorang juga rata2 sudah pada tidur setelah tarwih	['kok', 'ngumumin', 'hasil', 'pemilu', 'yang', 'amat', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'juga', 'rata2', 'sudah', 'pada', 'tidur', 'setelah', 'tarwih']
1447	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yang berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont	['dari', 'hasil', 'pemilu', 'kali', 'ini', 'kita', 'mengetahui', 'bahwa', 'mayoritas', 'penduduk', 'indonesia', '1', 'minim', 'literasi', '2', 'tidak', 'suka', 'perdebatan', 'yang', 'berdasarkan', 'teori', 'data', 'dan', 'fakta', '3', 'cenderung', 'duniawi', '4', 'sekarang', 'ya', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lagi', '5', 'lebih', 'suka', 'kekerasan', 'cont']

Fig 4. Tokenization Result

Next, we enter the stopword removal stage, the tweet text which is already in token form will be removed by words that have no meaning in the text. This process was assisted with library assistance from Sastrawi. This library provides a function that can provide a list of words in Indonesian that have no meaning. The stopword removal process will refer to the list of words provided by the library. Here are the results

Commented [HS13]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

	tokenized_text	text_after_stopword
1440	['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini', 'sudah', 'dari', 'awal', 'sejak', 'mk', 'dan', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'yang', 'kita', 'sudah', 'tau', 'di', 'mana', 'ujung', 'ceritanya', 'mari', 'kita', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']	['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam', 'awal', 'sejak', 'mk', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'tau', 'mana', 'ujung', 'ceritanya', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']
1441	['pemilu', 'hasil', 'bansos', 't']	['pemilu', 'hasil', 'bansos', 't']
1442	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'anda', 'ini', 'ketum', 'nasdem', 'saja', 'sebagai', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'sudah', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'anda', 'tidak', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'ketum', 'nasdem', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']
1443	['tetap', 'lawan', 'sampai', 'tolak', 'pemilu', 'curang', 'tidak', 'mengakui', 'pemimpin', 'dari', 'hasil', 'kecurangan', 'yang', 'tidak', 'beradab']	['tetap', 'lawan', 'tolak', 'pemilu', 'curang', 'mengakui', 'pemimpin', 'hasil', 'kecurangan', 'beradab']
1444	['saya', 'minta', 'kebijaksanaan', 'mui', 'yang', 'saat', 'ini', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'sebagai', 'umat', 'islam', 'kita', 'dituntut', 'untuk', 'adil', 'cc']	['minta', 'kebijaksanaan', 'mui', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'umat', 'islam', 'dituntut', 'adil', 'cc']
1446	['kok', 'ngumumin', 'hasil', 'pemilu', 'yang', 'amat', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'juga', 'rata', 'sudah', 'pada', 'tidur', 'setelah', 'tarwih']	['kok', 'ngumumin', 'hasil', 'pemilu', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'rata', 'tidur', 'tarwih']
1447	['dari', 'hasil', 'pemilu', 'kali', 'ini', 'kita', 'mengetahui', 'bahwa', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'tidak', 'suka', 'perdebatan', 'yang', 'berdasarkan', 'teori', 'data', 'dan', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'ya', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lagi', 'lebih', 'suka', 'kekerasan', 'cont']	['hasil', 'pemilu', 'kali', 'mengetahui', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'suka', 'perdebatan', 'berdasarkan', 'teori', 'data', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lebih', 'suka', 'kekerasan', 'cont']

Fig 5. Stopword removal Result

The final process is stemming. This stage functions to change words that have affixes into basic words. This process also uses the stemming function that already exists in the Sastrawi library. In this process, the text form will be returned from token to plain text. The following are some texts that have undergone the stemming process

Commented [HS14]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

	text_after_stopword	text_after_stemming
1440	['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam', 'awal', 'sejak', 'mk', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'tau', 'mana', 'ujung', 'ceritanya', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']	kaget hasil pilpres umum kpu malam awal sejak mk kpu lolos gibran jadi calon wakil prabowo jadi salah satu kandidat capres pemilu bak sinetron tau mana ujung cerita tunggu sinetron episode ikut nya
1441	['pemilu', 'hasil', 'bansos', 't']	pemilu hasil bansos t
1442	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'ketum', 'nasdem', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']	pemilu kok anggap perang kacau tum nasdem partai sokong utama paslon no nyata terima hasil pemilu mosok cepat cuci muka sono biar siuman
1443	['tetap', 'lawan', 'tolak', 'pemilu', 'curang', 'mengakui', 'pemimpin', 'hasil', 'kecurangan', 'beradab']	tetap lawan tolak pemilu curang aku pimpin hasil curang adab
1444	['minta', 'kebijaksanaan', 'mui', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'umat', 'islam', 'dituntut', 'adil', 'cc']	minta bijaksana mui tuai kyai kena hasil pemilu umat islam tuntut adil cc
1446	['kok', 'ngumumin', 'hasil', 'pemilu', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'rata', 'tidur', 'tarwih']	kok ngumumin hasil pemilu penting tengah malem bangett dahh orangorang rata tidur tarwih
1447	['hasil', 'pemilu', 'kali', 'mengetahui', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'suka', 'perdebatan', 'berdasarkan', 'teori', 'data', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lebih', 'suka', 'kekerasan', 'cont']	hasil pemilu kali tahu mayoritas duduk indonesia minim literasi suka debat dasar teori data fakta cenderung duniawi sekarang sekarang buat besok pikirinnya besok aja lebih suka keras cont

Fig 6. Stemming Result

DOI: doi.org/10.35970/jinita.vxixx.xx

C. LEXICON BASED LABELLING

At this stage, labeling of text data that has been previously processed will be carried out using a lexicon-based approach. At this stage, a dictionary has been prepared containing the words positive sentiment and also negative sentiment. After labeling is carried out, the data distribution for each positive, negative and neutral sentiment is as follows.

Commented [HS15]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

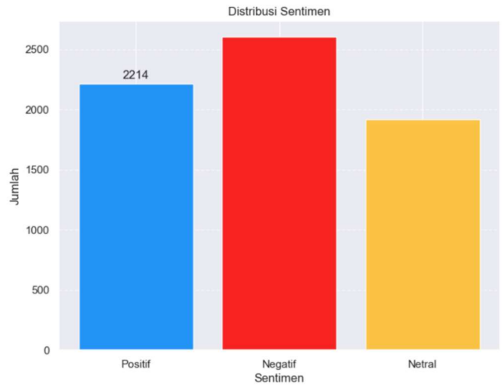


Fig 7. Sentiment Distribution

D. FEATURE EXTRACTION TF-IDF

In this process, feature extraction is carried out using the Term Frequency-Inverse Document Frequency (TF-IDF) method to convert text data into a numerical format that can be processed by a machine learning model. The results of the TF-IDF process produce 7256 features or words which have their respective weights in vector form. Figure 4.12 is an example of TF-IDF features and their weights in each document. The columns in the table represent each word in the entire sentence, while each row represents the sequence of the document or text

Commented [HS16]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

	bansos	curang	indonesia	mahkamah	pemilu	pilpres	prabowo	tolak
2993	0.000000	0.278981	0.169016	0.000000	0.126922	0.000000	0.000000	0.000000
2994	0.000000	0.000000	0.000000	0.168127	0.000000	0.121763	0.000000	0.000000
2995	0.000000	0.000000	0.000000	0.205216	0.000000	0.074312	0.000000	0.000000
2996	0.000000	0.000000	0.000000	0.000000	0.000000	0.124070	0.196398	0.000000
2997	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.165893	0.000000
2998	0.000000	0.000000	0.186157	0.126858	0.000000	0.091874	0.000000	0.000000
2999	0.000000	0.000000	0.000000	0.000000	0.109154	0.000000	0.227116	0.000000

Fig 8. Word in the entire sentence

E. STACKING MODELLING

At this stage, a sentiment analysis model will be built using the ensemble learning stacking method involving three algorithms as the base model, namely Naive Bayes, Random Forest, and Support Vector Machine, as well as Random Forest as a meta model. The data used in creating this model is divided in a ratio of 80:20, where 80% of the data is used for training and 20% for testing. The result is 5389 training data and 1348 testing data. The following is sample data

Commented [HS17]: In this section, it is necessary to mention the images used as references in the explanatory sentences.

	text_after_stemming	sentiment
150	hiv indonesia raya tingkat azab prararo nepo baby pemenang pemilu	Negatif
151	terus lampias marah presiden kalah pilpres hasil resmi kpu rubah apa presiden wakil presiden pilih sah	Netral
152	paman mahkamah konstitusi mbatalin hasil pemilu wakanda wakanda	Negatif
153	tidak banjir pesisir utara minggu minggu ganjar duluan terjun asih bantu presiden capres pemenang pemilu tidak nilai sombong rakyat	Negatif
154	silaturahmi maksud politisasi terima hasil pemilu	Positif
155	kuat argumen mahkamah konstitusi tidak rubah hasil pilpres gandeng pilpres politikus malin kundang anak mantu cundang	Netral
156	temu tidak hak tuan rumah anies tolak hak mas gibran temu anies sosok anies diri diri sengketa hasil pilpres belum selesai	Negatif
157	kpu umum pemenang pilpres pasang prabowo gibran menang jokowi versi secc akun	Positif
158	ambil contoh pp muhammadiyah sikap hasil pemilu dewasa sabar	Positif
159	bangga juara pileg tidak terima hasil pilpres otak kerdil	Netral

Fig 9. Training Data Sample

	text_after_stemming	sentiment
150	agam hasil pemilu cermin agam masyarakat hargai	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	Negatif
152	mahkamah konstitusi pas rapat rekapitulasi manis tidak tau umum pilpres tau mahkamah konstitusi timses ngumpulin bukti tidak salah pakai advokat kurang	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	Positif
154	malam ngobrolin hasil pemilu pajak pacar gara rekonsiliasi israel	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	Positif
156	surya paloh terima hasil pemilu sahabat presiden	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	Positif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	Negatif
159	tiktok ribut hasil pemilu kali ribut hasil pemilu ribut kpop sosmed irl ngantuk kerja stngh	Netral

Fig 10. Testing Data Sample

After the data is shared, each base model will be trained using the training data and will later produce predictions on the test data. Through the predict_proba function, each model will provide a probability for whether the data is labeled positive, negative or neutral. The class or label that has the highest probability will be used as the final prediction of the model

Commented [HS18]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

```
# Membuat meta-feature untuk training data
nb_train_pred = nb.predict_proba(X_train_tfidf)
rf_train_pred = rf.predict_proba(X_train_tfidf)
svm_train_pred = svm.predict_proba(X_train_tfidf)
meta_features_train = np.concatenate([nb_train_pred, rf_train_pred, svm_train_pred], axis=1)

# Membuat meta-feature untuk test data
nb_test_pred_proba = nb.predict_proba(X_test_tfidf)
rf_test_pred_proba = rf.predict_proba(X_test_tfidf)
svm_test_pred_proba = svm.predict_proba(X_test_tfidf)
meta_features_test = np.concatenate([nb_test_pred_proba, rf_test_pred_proba, svm_test_pred_proba], axis=1)

# Inisialisasi dan training meta learner
meta_learner = RandomForestClassifier(n_estimators=100, random_state=42)
meta_learner.fit(meta_features_train, y_train)

# Prediksi akhir dengan meta learner pada test data
final_predictions = meta_learner.predict(meta_features_test)
```

Fig 11. Stacking Code

Figure 2 shows the prediction results and the class probability of each model against the test data. The order of classes 0, 1 and 2 in the table shows the negative, neutral and positive classes

Commented [HS19]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

DOI: doi.org/10.35970/jinita.vxixx.xx

	text	nb_class0	nb_class1	nb_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.016227	0.027230	0.956543	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.622731	0.188593	0.188675	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.348763	0.402374	0.248863	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.360907	0.252019	0.387073	Positif
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.185071	0.594748	0.220181	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.300622	0.151513	0.547865	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.015130	0.032929	0.951941	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.564754	0.209396	0.225850	Negatif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.620861	0.141082	0.238057	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.427569	0.341746	0.230685	Negatif

Fig 12. Naïve Bayes Probability Result

	text	rf_class0	rf_class1	rf_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.081357	0.262206	0.656437	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.548884	0.226526	0.224590	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.129437	0.766280	0.104283	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.102069	0.206751	0.691180	Positif
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.455056	0.416384	0.128561	Negatif
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.277810	0.125952	0.596238	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.090937	0.169992	0.739071	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.314492	0.410989	0.274519	Netral
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.411056	0.261349	0.327595	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.675032	0.170857	0.154111	Negatif

Fig 13. Random Forest Probability Result

	text	svm_class0	svm_class1	svm_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.015999	0.139643	0.844358	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.742923	0.206424	0.050653	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.134347	0.786524	0.079129	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.023798	0.508969	0.467233	Netral
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.138457	0.789167	0.072376	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.002719	0.012429	0.984851	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.000480	0.015403	0.984117	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.411272	0.226814	0.361914	Negatif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.515261	0.302810	0.181929	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.461814	0.378116	0.160070	Negatif

Fig 14. SVM Probability Result

After all base models have their own predictions, the prediction results will be combined and used as features of the meta model. So the meta model will carry out training and testing data using these new features. The following is an example of a feature that will be used by the meta model to make final predictions

Commented [HS20]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

DOI: doi.org/10.35970/jinita.vxixx.xx

	nb_class0	nb_class1	nb_class2	rf_class0	rf_class1	rf_class2	svm_class0	svm_class1	svm_class2	predict
150	0.016227	0.027230	0.956543	0.081357	0.262206	0.656437	0.015628	0.135144	0.849228	Positif
151	0.622731	0.188593	0.188675	0.548884	0.226526	0.224590	0.742093	0.207783	0.050124	Negatif
152	0.348763	0.402374	0.248863	0.129437	0.766280	0.104283	0.137193	0.784752	0.078055	Netral
153	0.360907	0.252019	0.387073	0.102069	0.206751	0.691180	0.023476	0.507070	0.469453	Positif
154	0.185071	0.594748	0.220181	0.455056	0.416384	0.128561	0.141315	0.787167	0.071518	Negatif
155	0.300622	0.151513	0.547865	0.277810	0.125952	0.596238	0.002552	0.011500	0.985948	Positif
156	0.015130	0.032929	0.951941	0.090937	0.169992	0.739071	0.000436	0.014297	0.985267	Positif
157	0.564754	0.209396	0.225850	0.314492	0.410989	0.274519	0.410736	0.224927	0.364337	Netral
158	0.620861	0.141082	0.238057	0.411056	0.261349	0.327595	0.515909	0.302254	0.181837	Positif
159	0.427569	0.341746	0.230685	0.675032	0.170857	0.154111	0.462122	0.378279	0.159599	Negatif

Fig 15. Stacking Dataset And Result

F. EVALUATION

At this stage, all models that have been trained and have produced predictions will be evaluated to see their performance using the Confusion Matrix. The evaluation matrix used includes accuracy, precision recall, and F1-score. Each matrix will be calculated using the following formula

The Confusion Matrix for the Naïve Bayes model shows that the model succeeded in predicting correctly (True Positive) 323 data with positive sentiment, 105 data with neutral sentiment, 473 data with negative sentiment

Commented [HS21]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

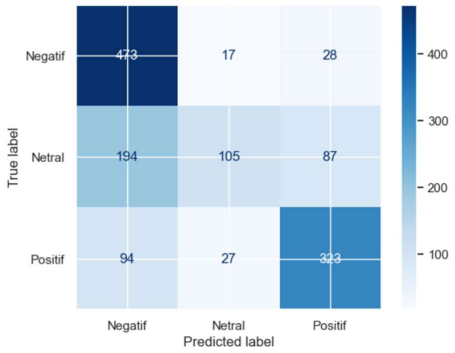


Fig 16. Naïve bayes Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the Naïve Bayes model

$$Accuracy = \frac{323 + 105 + 403}{1348} = \frac{829}{1348} = 0.6684$$

The accuracy of the Naïve Bayes model is 0.6684

$$Precission\ Positif = \frac{323}{323 + 115} = \frac{323}{438} = 0.7374$$

$$Precission\ Netral = \frac{105}{105 + 44} = \frac{105}{149} = 0{,}7047$$

$$Precission\ Negatif = \frac{473}{473 + 288} = \frac{473}{761} = 0{,}6216$$

$$Precission_{weighted} = \frac{(444 \times 0{,}7347) + (386 \times 0{,}7047) + (518 \times 0{,}6216)}{444 + 386 + 518} = 0.6835$$

The precission of the Naïve Bayes model is 0.6835

$$Recall\ Positif = \frac{323}{323 + 121} = \frac{323}{444} = 0{,}7275$$

$$Recall\ Netral = \frac{105}{105 + 281} = \frac{105}{386} = 0.272$$

$$Recall\ Negatif = \frac{473}{473+45} = \frac{473}{518} = 0.9131$$

$$Recall_{weighted} = \frac{(444 \times 0{,}7275) + (386 \times 0.272) + (518 \times 0.9131)}{444+386+518} = 0,6684$$

The recall of the Naïve Bayes model is 0.6684

$$F-1\ Score\ Positif = 2\,x\,\frac{(0{,}7374 \times 0{,}7275)}{(0{,}7374+0{,}7275)} = 0{,}7324$$

$$F-1\ Score\ Netral = 2\,x\,\frac{(0{,}7047 \times 0{,}272)}{(0{,}7047+0{,}272)} = 0{,}3925$$

$$F-1\ Score\ Negatif = 2\,x\,\frac{(0{,}6216 \times 0{,}9131)}{(0{,}6216+0{,}9131)} = 0{,}7397$$

$$F-1 Score_{weighted} = \frac{(444 \times 0{,}7324) + (386 \times 0.3925) + (518 \times 0.7397)}{444+386+518} = 0,6379$$

The F-1 Score of the Naïve Bayes model is 0.6379

The Confusion Matrix for the Random Forest model shows that the model succeeded in predicting correctly (True Positive) 348 data with positive sentiment, 232 data with neutral sentiment, 428 data with negative sentiment

Commented [HS22]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

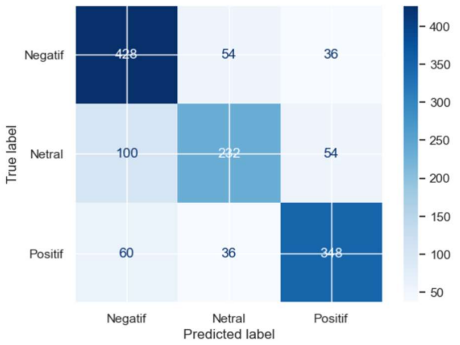


Fig 17. Random Forest Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the Random Forest model

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{348+232+428}{1348} = \frac{1008}{1348} = 0.7478$$

The accuracy of the Random Forest model is 0.7478

$$Precission\ Positif = \frac{348}{348+90} = \frac{348}{438} = 0.7945$$

$$Precission\ Netral = \frac{232}{232+90} = \frac{232}{222} = 0{,}7205$$

$$Precission\ Negatif = \frac{428}{428+160} = \frac{428}{588} = 0{,}7279$$

$$Precission_{weighted} = \frac{(444 \times 0{,}7945) + (386 \times 0{,}7205) + (518 \times 0{,}7279)}{444+386+518} = 0.7477$$

The precission of the Random Forest model is 0.7477

$$Recall\ Positif = \frac{348}{348+96} = \frac{348}{444} = 0{,}7838$$

$$Recall\ Netral = \frac{232}{232+154} = \frac{232}{386} = 0.601$$

$$Recall\ Negatif = \frac{428}{428+90} = \frac{428}{518} = 0.8263$$

$$Recall_{weighted} = \frac{(444 \times 0{,}7838) + (386 \times 0.601) + (518 \times 0.8263)}{444+386+518} = 0{,}7478$$

The recall of the Random Forest model is 0.7478

$$F-1\ Score\ Positif = 2\,x\ \frac{(0{,}7945 \times 0{,}7838)}{(0{,}7945+0{,}7838)} = 0{,}7891$$

$$F-1\ Score\ Netral = 2\,x\ \frac{(0{,}7205 \times 0{,}601)}{(0{,}7205+0{,}601)} = 0{,}6553$$

$$F-1\ Score\ Negatif = 2\,x\ \frac{(0{,}7279 \times 0{,}8263)}{(0{,}7279+0{,}8263)} = 0{,}774$$

$$F-1 Score_{weighte} = \frac{(444 \times 0{,}6718) + (386 \times 0.5771) + (518 \times 0.7157)}{444+386+518} = 0{,}745$$

The F-1 Score of the Random Forest model is 0.745

The Confusion Matrix for the SVM model shows that the model succeeded in predicting correctly (True Positive) 370 data with positive sentiment, 252 data with neutral sentiment, 426 data with negative sentiment

Commented [HS23]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

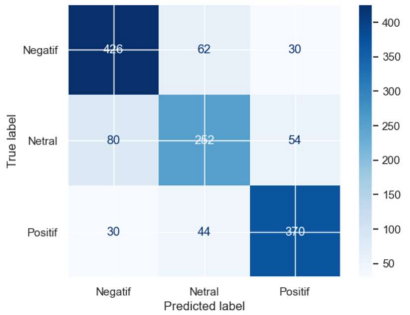


Fig 18. SVM Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the SVM model

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{370+252+426}{13512} = \frac{1048}{1351} = 0.7774$$

The accuracy of the SVM model is 0.7774

$$Precission\ Positif = \frac{370}{370+84} = \frac{370}{454} = 0.815$$

$$Precission\ Netral = \frac{252}{252+106} = \frac{252}{358} = 0{,}7039$$

$$Precission\ Negatif = \frac{426}{426+110} = \frac{426}{536} = 0{,}7984$$

$$Precission_{weighted} = \frac{(444 \times 0{,}815) + (386 \times 0{,}7039) + (518 \times 0{,}7984)}{444+386+518} = 0.7754$$

The precission of the naïve Bayes model is 0.7754

$$Recall\ Positif = \frac{370}{370+74} = \frac{370}{444} = 0{,}8333$$

$$Recall\ Netral = \frac{252}{252+134} = \frac{252}{386} = 0.6528$$

$$Recall\ Negatif = \frac{426}{426+92} = \frac{426}{518} = 0.8224$$

$$Recall_{weighted} = \frac{(444 \times 0{,}8333) + (386 \times 0.6528) + (518 \times 0.8224)}{444+386+518} = 0,7774$$

The recall of the SVM model is 0.7774

$$F-1\ Score\ Positif = 2\,x\,\frac{(0{,}815 \times 0{,}8333)}{(0{,}815+0{,}8333)} = 0{,}824$$

$$F-1\ Score\ Netral = 2\,x\,\frac{(0{,}7039 \times 0{,}6528)}{(0{,}7039+0{,}6528)} = 0{,}6774$$

$$F-1\ Score\ Negatif = 2\,x\,\frac{(0{,}7984 \times 0{,}8224)}{(0{,}7984+0{,}8224)} = 0{,}8084$$

$$F-1Score_{weighted} = \frac{(444 \times 0{,}824) + (386 \times 0.6774) + (518 \times 0.8084)}{444+386+518} = 0,8084$$

The F-1 Score of the SVM model is 0.8084

The Confusion Matrix for the stacking model shows that the model succeeded in predicting correctly (True Positive) 396 data with positive sentiment, 256 data with neutral sentiment, 447 data with negative sentiment.

Commented [HS24]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

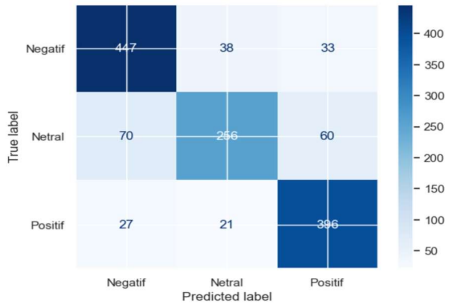


Fig 19. Stacking Model's Confussion Matrix

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{396+256+447}{13512} = \frac{1099}{1351} = 0.8153$$

Below are calculations to find the accuracy, precision, recall, and F1-score of the Stacking model

$$Precission\ Positif = \frac{396}{396+93} = \frac{396}{489} = 0.8089$$

$$Precission\ Netral = \frac{256}{256+59} = \frac{256}{315} = 0.8127$$

$$Precission\ Negatif = \frac{447}{447+97} = \frac{447}{544} = 0{,}8127$$

$$Precission_{weighted} = \frac{(444 \times 0{,}8089) + (386 \times 0{,}8127) + (518 \times 0{,}8127)}{444+386+518} = 0.8152$$

The accuracy of the Stacking model is 0.8152

$$Recall\ Positif = \frac{396}{396+48} = \frac{396}{444} = 0{,}8919$$

$$Recall\ Netral = \frac{256}{256+130} = \frac{256}{386} = 0.6632$$

$$Recall\ Negatif = \frac{447}{447+71} = \frac{447}{518} = 0.8629$$

$$Recall_{weighted} = \frac{(444 \times 0{,}8919) + (386 \times 0.6632) + (518 \times 0.8629)}{444+386+518} = 0,8153$$

The recall of the Stacking model is 0.8153

$$F-1\ Score\ Positif\ =\ 2\ x\ \frac{(0{,}8098 \times 0{,}8919)}{(0{,}8098 + 0{,}8919)} = 0{,}8489$$

$$F-1\ Score\ Netral\ =\ 2\ x\ \frac{(0{,}8127 \times 0{,}6632)}{(0{,}8127 + 0{,}6632)} = 0{,}7304$$

$$F-1\ Score\ Negatif\ =\ 2\ x\ \frac{(0{,}8217 \times 0{,}8629)}{(0{,}8217 + 0{,}8629)} = 0{,}8418$$

$$F-1 Score_{weighted} = \frac{(444 \times 0{,}6718) + (386 \times 0.5771) + (518 \times 0.7157)}{444 + 386 + 518} = 0,8122$$

The F-1 Score of the Stacking model is 0.8122

The Result of all model can be seen in the table below

Table 3. Results Of Model

Model	Accuracy	Precission	Recall	F-1 Score
Naïve Bayes	0.6684	0.6835	0,6684	0,6379
Support Vector Machine	0.7774	0.7754	0,7774	0,776
Random Forest	0.7478	0.7477	0,7478	0,745
Ensemble Stacking (RF)	0.8153	0.8152	0,8153	0,8122

Commented [HS25]: In the table format section, it needs to be adjusted to the provisions in the JINITA journal.

4. CONCLUSION

As a result of this experiment, an ensemble learning stacking model was formed with several different base models, namely the SVM, Random Forest and Naïve Bayes algorithms. Each model carries out training and predictions on sentiment analysis data. The results, starting from the lowest, are the Naïve Bayes algorithm with an accuracy of 66.84%, followed by Random Forest with an accuracy of 74.78%, and the highest is SVM with an accuracy of 77.74%. The results of the three base models are compiled and used as input for a meta model that uses the Random Forest algorithm. The results show that the stacking ensemble method applied produces better accuracy than a single classifier, namely 81.53%. Thus, it can be concluded that ensemble learning stacking with the SVM, Random Forest and Naïve Bayes base models as well as meta models using Random Forest can increase the accuracy of sentiment analysis models on unstructured text data

Commented [HS26]: This section needs an explanatory sentence that explains the impact of ensemble learning as a contribution in this research.

REFERENCES

[1] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, 2023, doi: 10.1016/j.jksuci.2023.01.014.

[2] F. Matloob *et al.*, "Software defect prediction using ensemble learning: A systematic literature review," *IEEE Access*, vol. 9, pp. 98754–98771, 2021, doi: 10.1109/ACCESS.2021.3095559.

[3] Y. Görmez, Y. E. Işık, M. Temiz, and Z. Aydın, "FBSEM: A Novel Feature-Based Stacked Ensemble Method for Sentiment Analysis," *Int. J. Inf. Technol. Comput. Sci.*, vol. 12, no. 6, pp. 11–22, 2020, doi: 10.5815/ijitcs.2020.06.02.

[4] P. Thiengburanathum and P. Charoenkwan, "SETAR: Stacking Ensemble Learning for Thai Sentiment Analysis Using RoBERTa and Hybrid Feature Representation," *IEEE Access*, vol. 11, no. August, pp. 92822–92837, 2023, doi: 10.1109/ACCESS.2023.3308951.

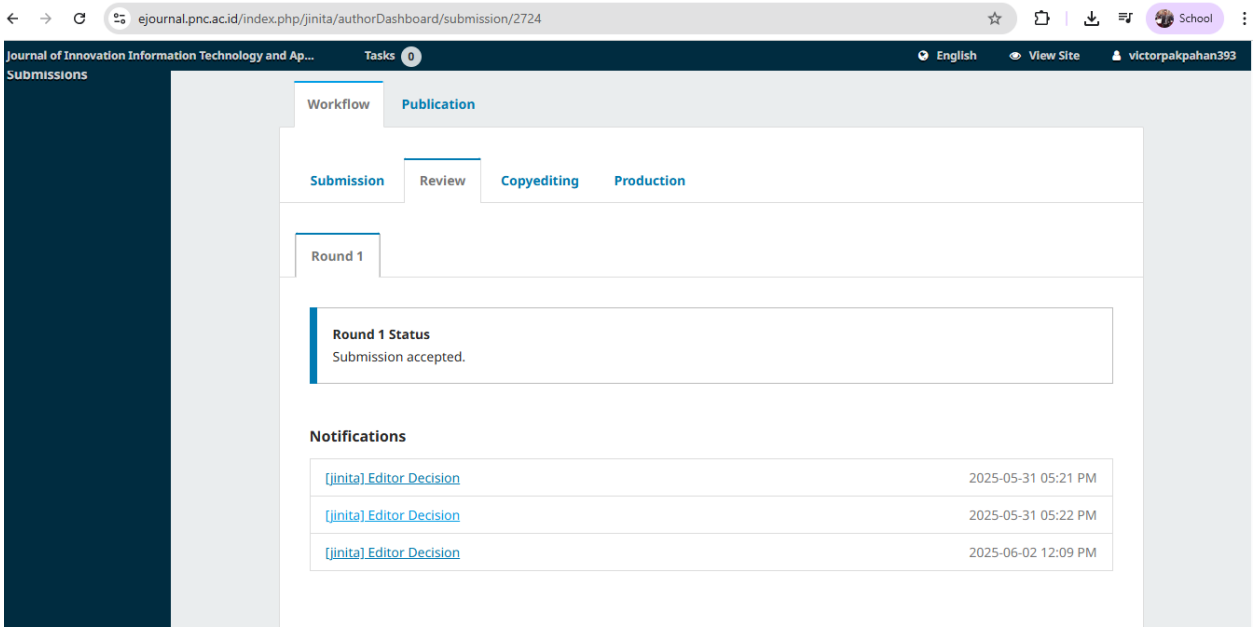
[5] J. Gu, S. Liu, Z. Zhou, S. R. Chalov, and Q. Zhuang, "A Stacking Ensemble Learning Model for Monthly Rainfall Prediction in the Taihu Basin, China," *Water (Switzerland)*, vol. 14, no. 3, 2022, doi: 10.3390/w14030492.

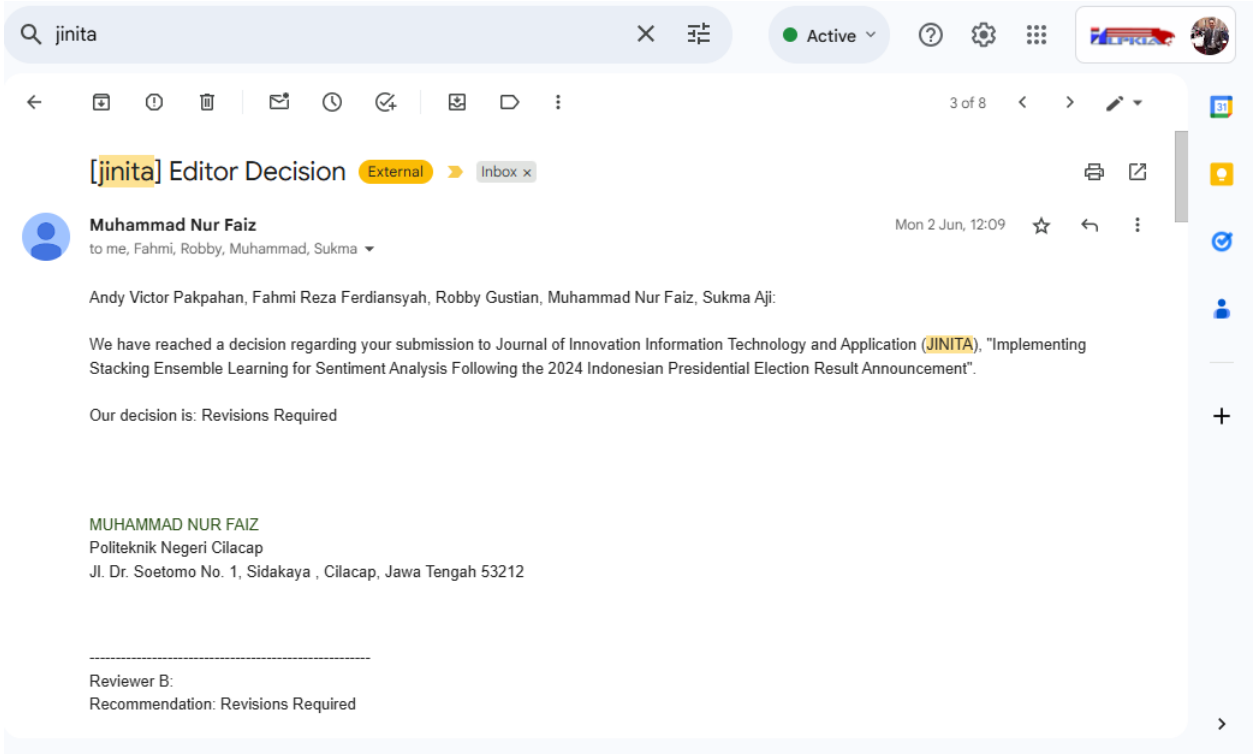
[6] S. A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, "Stacking Strong Ensembles of Classifiers," in *IFIP Advances in Information and Communication Technology*, 2019. doi: 10.1007/978-3-030-19823-7_46.

[7] J. Dou *et al.*, "Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan," *Landslides*, vol. 17, no.

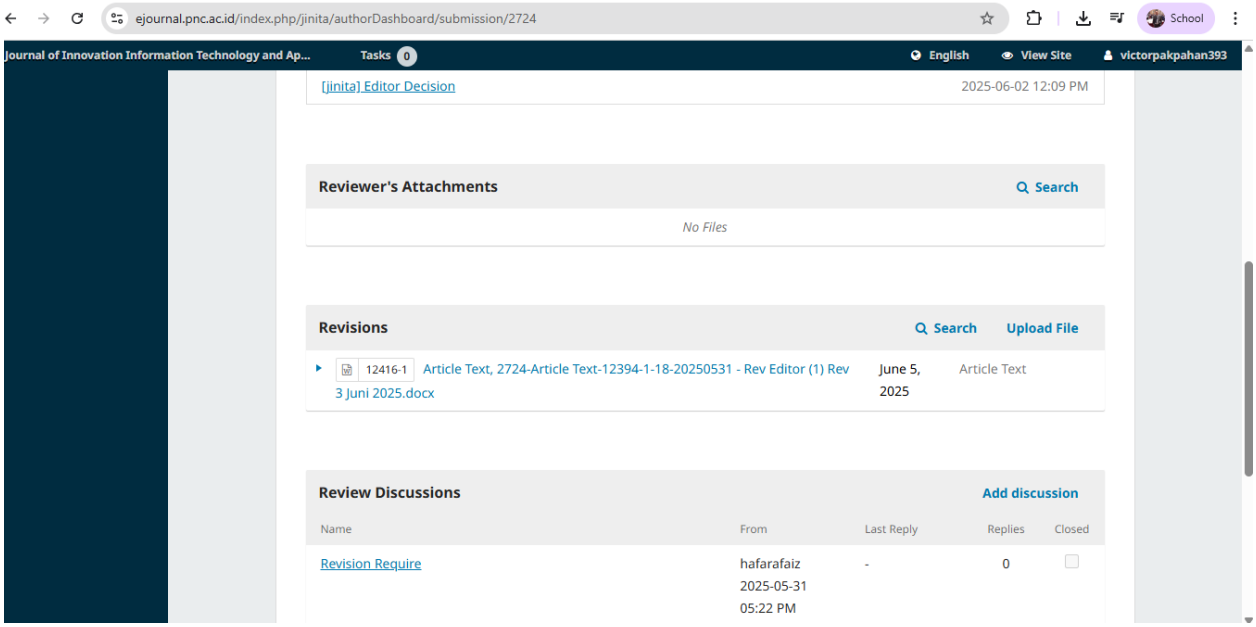
3, 2020, doi: 10.1007/s10346-019-01286-5.
[8] I. D. Mienye and Y. Sun, “A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects,” *IEEE Access*, vol. 10, no. September, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
[9] M. B. Alfarazi, M. ’Ariful Furqon, and H. Soepandi, “Sentiment Analysis of User Reviews for the Sister For Student Application Using Gaussian Naive Bayes and N-Gram,” *J. Innov. Inf. Technol. Appl.*, pp. 101–108, 2024.
[10] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artif. Intell. Rev.*, vol. 55, no. 7, 2022, doi: 10.1007/s10462-022-10144-1.
[11] A. M. Iddrisu, S. Mensah, F. Boafo, G. R. Yeluripati, and P. Kudjo, “A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 2, 2023, doi: 10.1016/j.jjimei.2023.100180.
[12] R. Michaela Denise Gonzales and C. A. Hargreaves, “How can we use artificial intelligence for stock recommendation and risk management? A proposed decision support system,” *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, 2022, doi: 10.1016/j.jjimei.2022.100130.
[13] A. Tripathi, T. Goswami, S. K. Trivedi, and R. D. Sharma, “A multi class random forest (MCRF) model for classification of small plant peptides,” *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100029, 2021, doi: 10.1016/j.jjimei.2021.100029.
[14] M. Y. Aldean, P. Paradise, and N. A. Setya Nugraha, “Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 di Twitter Menggunakan Metode Random Forest Classifier (Studi Kasus: Vaksin Sinovac),” *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 4, no. 2, pp. 64–72, 2022, doi: 10.20895/inista.v4i2.575.
[15] A. Syafrianto, “Perbandingan Algoritma Naïve Bayes dan Decision Tree Pada Sentimen Analisis,” *Indones. J. Comput. Sci. Res.*, vol. 1, no. 2, pp. 1–15, 2022, doi: 10.59095/ijcsr.v1i2.11.
[16] R. Nuraeni, A. Sudiarjo, and R. Rizal, “Perbandingan Algoritma Naïve Bayes Classifier dan Algoritma Decision Tree untuk Analisa Sistem Klasifikasi Judul Skripsi,” *Innov. Res. Informatics*, vol. 3, no. 1, pp. 26–31, 2021, doi: 10.37058/innovatics.v3i1.2976.
[17] S. H. Ramadhani and M. I. Wahyudin, “Analisis Sentimen Terhadap Vaksinasi Astra Zeneca pada Twitter Menggunakan Metode Naïve Bayes dan K-NN,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 4, 2022, doi: 10.35870/jtik.v6i4.530.
[18] M. Asgari, W. Yang, and M. Farnaghi, “Spatiotemporal data partitioning for distributed random forest algorithm: Air quality prediction using imbalanced big spatiotemporal data on spark distributed framework,” *Environ. Technol. Innov.*, vol. 27, p. 102776, 2022, doi: 10.1016/j.eti.2022.102776.
[19] G. Meena, K. K. Mohbey, and S. Kumar, “Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 1, p. 100174, 2023, doi: 10.1016/j.jjimei.2023.100174.
[20] K. L. Tan, C. P. Lee, and K. M. Lim, “A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research,” *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074550.

5. JINITA Editor Decision – Revisions Required – Reminder





6. Unggah Revisi Artikel



Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Sentiment Analysis Using Stacking Ensemble After 2024 Indonesian Election Results

ARTICLE INFO

Article history:
Received 24 January 2020
Revised 30 April 2020
Accepted 2 December 2020
Available online xxx

Keywords:
Ensemble Learning,
Stacking,
Sentiment Analysis,
Data Mining,
Machine Learning

IEEE style in citing this article:
Jinita and J. Jinita, "Article Title," *Journal of Innovation Information Technology and Application*, vol. 4, no. 1, pp. 1-10, 2022. [Fill citation heading]

ABSTRACT

Sentiment analysis is a text processing technique aimed at identifying opinions and emotions within a sentence. Machine learning is commonly applied in this area, with algorithms such as Naïve Bayes, Support Vector Machine (SVM), and Random Forest being frequently used. However, achieving optimal accuracy remains a challenge, particularly when dealing with unstructured text data, such as content from social media platforms. This research seeks to improve sentiment analysis performance by implementing a stacking ensemble learning approach, which combines the predictive strengths of several base models. The base models selected for this study are Naïve Bayes, SVM, and Random Forest, while Random Forest also serves as the meta-model to generate final predictions.

The study focuses on sentiment analysis in a specific context—public opinion following the announcement of the Indonesian presidential election results in 2024. The dataset comprises 6,737 tweets collected from the X platform using web scraping techniques in 2024. Results show that individual models achieved varying levels of accuracy: Naïve Bayes at 66.84%, SVM at 77.74%, and Random Forest at 74.78%. In contrast, the stacking ensemble model achieved a significantly higher accuracy of 81.53%. This improvement highlights the effectiveness of ensemble learning in integrating different algorithmic perspectives to enhance predictive performance. By leveraging the complementary strengths of each base model, stacking not only boosts accuracy but also increases model robustness, making it highly suitable for real-world sentiment analysis applications that involve noisy and informal textual data from social media.

1. INTRODUCTION

The Ensemble Learning is a method in machine learning that combines several models to create a new model that is stronger than and has superior performance compared to when the algorithms are used individually [1], [2]. There are several ensemble learning techniques such as bagging, stacking, averaging and boosting, each technique is distinguished by how the model is trained and combined [1].

Stacking is an ensemble learning technique that works by combining the results of several different base-models. Each base-model will learn and have its own prediction results, after that a final model will be created which will combine the prediction results of all the base-models which is called a meta-model [3], [4]. The Stacking technique is based on the idea that each basic model has its own advantages and disadvantages [5]. By combining predictions from different base-models, the resulting meta-model can learn and balance these advantages and disadvantages appropriately, so that the overall performance of the stacking model can exceed the performance of any individual model and makes it a fairly good technique for improving predictive power of the classifier [6], [7]. This is the advantage of the stacking technique compared to other ensemble learning techniques and makes stacking a suitable technique for creating models for processing quite complex data such as sentiment analysis [8]

Sentiment analysis is the process of understanding, extracting and processing textual data automatically to obtain information on opinions, feelings and emotions contained in a sentence [9]. Sentiment analysis aims to understand a person's level of satisfaction and dissatisfaction with a service or product, as well as

*) Corresponding Author : abang@lpkia.ac.id

understanding public perceptions regarding a person's agreement and disagreement with a particular topic [10]

Sentiment analysis is generally made using classification algorithm models such as Support Vector Machine (SVM), Decission Tree, K-Nearest-Neighbor (KNN), Naïve Bayes, Random Forest, etc [11], [12]. Several classification algorithms have been used in several previous studies regarding sentiment analysis carried out on opinions taken from social media X or Twitter in Indonesian and each algorithm has different accuracy [11]. Comparing the SVM algorithm with other algorithms such as Naïve Bayes, Decission Tree and KNN in sentiment analysis with different cases or topics, the result is that SVM accuracy is better when compared to other algorithms. Even though Naïve Bayes is not superior in accuracy to the SVM algorithm, if we refer to research conducted Naïve Bayes still has better accuracy results when compared to the Decision Tree and KNN algorithms [13]. Then, if we refer to research which compares the Random Forest algorithm with other algorithms such as Naïve Bayes, KNN, Decission Tree and Logistic Regression, it can be seen that Random Forest produces better accuracy than other algorithms including SVM [14]. Other research that shows that Random Forest is superior to SVM is research From these studies, it can be seen that the Random Forest, SVM and Naïve Bayes algorithms are some of the algorithms with the best accuracy in terms of sentiment analysis.

Even so, sentiment analysis is not an easy task to do. The complexity of language and variations in human expressions in various sentences make sentiment analysis a challenge [15], [16]. Building a model that can produce accuracy and good performance is also a challenge in sentiment analysis [17] especially sentiment analysis of unstructured text, for example data taken from social media such as X or Twitter has its own challenges because the language used is usually not appropriate. standard words, involving abbreviations, as well as words that are not in the dictionary, thus affecting accuracy[18], [19]. So the accuracy of the sentiment analysis model can still be improved with the help of other methods, for example by using the ensemble stacking method.

Based on the description above, a sentiment analysis model will be built using the ensemble learning stacking method, with the aim of increasing the accuracy of the model in sentiment analysis on unstructured text, which in this case is data collected via the social media platform [10], [20]. This sentiment is the public's opinion regarding the results of the 2024 Indonesian Presidential Election. We propose to use Naive Bayes, Random Forest and Support Vector Machine as base models and Random Forest as a meta model because these models are suitable for sentiment analysis and have been widely used by previous researchers. Besides that, these models also have different characteristics.

2. METHOD

The research flow presented in this experiment outlines the structured steps taken to achieve the objectives of the study. It begins with the identification of the problem, which serves as the foundation for formulating the research questions and determining the appropriate methodology. This initial phase is crucial to ensure that the research direction is clear and aligned with the intended goals.

Following problem identification, the flow continues through stages such as data collection, analysis, and interpretation. Each step is interconnected, allowing the process to build logically upon the previous one. This structured approach not only helps in maintaining the consistency of the study but also enhances the reliability and validity of the results obtained.

Figure 1 provides a visual representation of this research flow. It serves as a guide to understand how the experiment was conducted from start to finish. By presenting the process in a flowchart format, it becomes easier to grasp the overall methodology and appreciate the systematic effort involved in reaching the research conclusions.

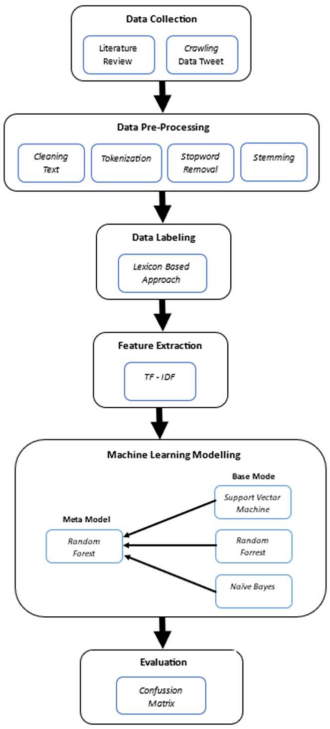


Fig 1. Flow of Research

A. DATA COLLECTION

Based on the description above, a sentiment analysis model will be built using the ensemble learning stacking method, with the aim of increasing

B. DATA PRE-PROCESSING

This process includes a series of steps to prepare the data before creating a sentiment analysis model. Stages that will be carried out in the process pre-processing data is as follows:

1) CLEANING TEXT

At this stage, text data will be cleaned has been collected from scrapping results so that the text can be made easier processed by the next stage. Data cleaning includes several processes such as deleting numbers and symbols, changing text to lowercase and also normalize the text or change each word in a sentence becomes standard or normal form for omission non-standard words, abbreviations, slang words, typo words etc. Text normalization This is done by referring to the dictionary provided which contains non-standard words and actual standard words.

2) TOKENIZATION

At this stage, every text data has been cleaned will be convert into small parts of each word in sentences called tokens. For example, sentence “Indonesia Lebih Maju” will be converted to ["Indonesia", "lebih", "maju"]

3) STOPWORD REMOVAL

At this stage, any common words are not make significant contributions to the meaning of the text will be removed. The stopword dictionary will be taken from a library that has provided a list the stop words are

Sastrawi. Some examples of words included in the stopword and will be deleted are like “yang”, “dan”, “di”, “adalah”.

4) STEMMING

At this stage, every word in the text will be changed be the basic word. Words with the same ending or words those with affixes will be changed to the basic form.

C. DATA LABELING

The method that will be used for data labeling is Lexicon Based. Lexicon based Approach can be used to create labeled training datasets for sentiment analysis machine learning algorithms that require labels at the start of his training [23]. The idea behind the lexicon based approach is that the meaning of a text is greatly influenced by the polarity of the words and phrases inside. This includes words such as adjectives, adverbs, nouns, verbs, as well as phrases and sentences that contain them [24]. This approach makes use of a dictionary or list of words with predefined sentiment labels. Any data will be carried out Check the total score of positive words and negative words. If the word score positive exceeds negative scores, then the label is positive, and vice versa then the label is negative. However, if the score is the same or 0, then the label will be neutral.

D. FEATURE EXTRACTION

In this process, feature extraction will be carried out using the Term Frequency-Inverse Document Frequency (TF-IDF). At this stage, every tweet will be represented as a numerical feature vector, where each component The vector will represent the weight of each word in the existing word dictionary. This weight calculated based on the frequency of occurrence of words in tweets (TF) and inverse proportional to the occurrence of the word in the entire collection of tweets (IDF). This feature extraction process aims to change the tweet text into a numerical representation that can be used by the model to perform further analysis. The formulas used for calculating Term Frequency-Inverse Document Frequency (TF-IDF) are as follows

$$TF(t,d) = \frac{number\ of\ occurences\ of\ word\ in\ document}{total\ number\ if\ words\ in\ document} \quad (1)$$

$$IDF(t,D) = log\ log\ \left(\frac{N}{df(t,D)}\right) \quad (2)$$

$$TF - IDF(t,d) = TF(t,d)x\ IDF(t,D) \quad (3)$$

To implement the TF-IDF method effectively, it is essential to understand the meaning of each variable used in the formulas. Below are the definitions of the terms involved:

- N : Total number of documents in the collection
- df(t,D) : Number of documents in the collection containing term t
- TF(t,d) : Term Frequency of term t in document d
- IDF(t,D) : Inverse Document Frequency of term t in all documents D
- W(t,d) : Weight of term t in a document

These variables are used in the equations for TF, IDF, and the final TF-IDF score, which together represent the importance of a term within a specific document in relation to a corpus of documents.

E. STACKING MODELLING

At this stage a model will be built for sentiment analysis. Before that, The data will first be split into 2 parts, namely training data and testing data with a percentage of 80% training data and 20% test data. Then, it will be done training on each base model, namely naïve Bayes, support vector machine and random forest use data that has been split. Every base the model will generate predictions based on these features. As part of the stacking ensemble learning technique, the output from each base model is used as input for the meta model to generate the final prediction. This process is illustrated in Figure 2, which shows the stacking model architecture. Meanwhile, Table 1 describes the algorithm used, outlining the steps of training base models, collecting their prediction probabilities, and feeding them into the meta model. This structure allows the meta model to learn from multiple perspectives, improving overall prediction performance.

Commented [1]: What does this part mean? If explaining the formulas, add the necessary sentences first.

Commented [2]: In this section, it is necessary to mention Figure 2 and Table 1 as references. In addition, it is necessary to explain Table 1 also related to the algorithm used, at least 4-5 sentences containing the main sentence and explanatory sentences.

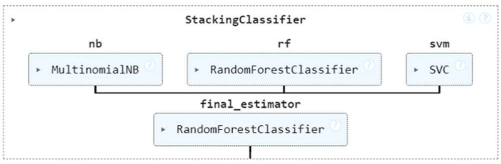


Fig 2. Stacking Model Illustration

Here is the algorithm for the stacking model that will be created

Table 1. Algortihm Stacking

Algorithm 1. Stacking
Input: X_train, y_train, x_test, base_models, meta_model
Output: prediction meta model
1. START
2. base_model_outputs_train = []
3. FOR model in base_models THEN
4. model.fit(X_train, y_train)
5. probas_train = model.predict_proba(X_train)
6. base_model_outputs_train.append(probas_train)
7. END FOR
meta_features_train = np.hstack(base_model_outputs_train)
meta_model.fit(meta_features_train, y_train)
8. base_model_outputs_test = []
FOR model in base_models THEN
9. probas_test = model.predict_proba(X_test)
10. base_model_outputs_test.append(probas_test)
11. END FOR
meta_features_test = np.hstack(base_model_outputs_test)
final_predictions = meta_model.predict(meta_features_test)
12. END

F. EVALUATION

Commented [3]: This section should be bolded.

After the model building process is complete, the next step is evaluate model performance using confusion matrix. Evaluation is carried out against each base model and meta model itself, so you can see the comparison of classification results between models. Because this sentiment analysis involves three classes—positive, negative, and neutral—the evaluation uses **weighted average** calculations. The metrics applied are **Accuracy (Equation 4)**, **Precision (Equation 5)**, **Recall (Equation 6)**, and **F1-Score (Equation 7)** to ensure fair assessment across all classes.

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} \quad (4)$$

$$Precision_{weighted} = \frac{\sum_{i=1}^{N} (Precision_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (5)$$

$$Recall_{weighted} = \frac{\sum_{i=1}^{N} (Recall_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (6)$$

$$F-1Score_{weighted} = \frac{\sum_{i=1}^{N} (F-1score_i\ x\ Total\ Data_i)}{\sum_{i=1}^{N} Total\ Data_i} \quad (7)$$

These evaluation metrics provide a comprehensive view of the model's ability to correctly classify sentiments across all classes. By using weighted averages, the metrics take into account the proportion of each class, ensuring that imbalanced class distributions do not bias the results. This is particularly important in multiclass classification problems where some classes may dominate. The use of these formulas allows for a fair comparison of performance between base models and the meta model.

3. RESULTS AND DISCUSSION

A. WEB SCRAPPING RESULT

The total data collected was 8094 tweets with details of each keyword are as follows.

Table 2. Data Collected by keyword.

Commented [4]: In the table format section, it needs to be adjusted to the provisions in the JINITA journal.

Keyword	Query Search	Result
Hasil pemilu	Hasil pemilu lang:id until:2024-04-30 since:2024-03-20 -filter:links -filter:repliess	3482 tweet
Hasil pemilu presiden	Hasil pemilu presiden lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	156 tweet
Hasil pilpres	Hasil pilpres lang:id until:2024-04-30 sinc e:2024-03-20 -filter:links -filter:replies	1983 tweet
Pemenang pemilu	Pemenang pemilu lang:id until:2024-04-3 0 since:2024-03-20 -filter:links -filter:rep liess	1000 tweet
Pemenang pilpres	Pemenang pilpres lang:id until:2024-04-3 0 since:2024-03-20 -filter:links -filter:rep liess	808 tweet
Pemenang presiden	Pemenang presiden lang:id until:2024-04 30 since:2024-03-20 -filter:links -filter:replies	274 tweet
Pengumuman pemilu	Pengumuman pemilu lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	221 tweet
Pengumuman pilpres	Pengumuman pilpres lang:id until:2024-0 4-30 since:2024-03-20 -filter:links -filter: repliess	170 tweet

B. DATA PRE-PROCESSING

The pre-processing stage is the first step in preparing the dataset by carrying out several stages, namely cleaning text, tokenization, stopword removal and stemming as well as deleting duplicate data. Data cleaning of scrapped tweet text includes several processes such as deleting mentions, deleting hashtags, deleting retweets, deleting URLs, deleting non-alphanumeric characters, deleting double spaces and transform the text into lowercase. Then, text normalization will be carried out to change words such as abbreviations, non-standard words, and slang words into normal and formal words. Finally, to avoid data duplication, tweet data that has the same or duplicate sentences will be deleted. So the final total of tweet data that will be used in the next stage until the end is 6737 tweets. The following results are based on the analysis shown in the reference image.

	lowered_text	normalized_text
1440	sudah tdk kaget dgn hasil pilpres yg di umumkan kpu malam ini sdh dr awal sejak mk dan kpu meloloskan gibram menjadi calon wakil prabowo menjadi salala satu kandidat capres pemilu bak sinetron yg kita sdh tau di mna ujung ceritanya mari kita tunggu sinetron episode berikut nya	sudah tidak kaget dengan hasil pilpres yang di umumkan kpu malam ini sudah dari awal sejak mk dan kpu meloloskan gibran menjadi calon wakil prabowo menjadi salah satu kandidat capres pemilu bak sinetron yang kita sudah tau di mana ujung ceritanya mari kita tunggu sinetron episode berikut nya
1441	pemilu hasil bansos 465 t	pemilu hasil bansos 465 t
1442	pemilu kok dianggap perang kacau ente ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok nte tidak cepat cuci muka sono biar siuman	pemilu kok dianggap perang kacau anda ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok anda tidak cepat cuci muka sono biar siuman
1443	tetap lawan amp tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yg tak beradab	tetap lawan sampai tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yang tidak beradab
1444	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc
1446	kok ngumumin hasil pemilu yg amat penting tengah malem bangett dahh orangorang juga rata2 udh pada tidur abis tarwih	kok ngumumin hasil pemilu yang amat penting tengah malem bangett dahh orangorang juga rata2 sudah pada tidur abis tarwih
1447	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yg berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yang berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont

Fig 3. Cleaning Text Result

After the normalization stage is complete, the next step in the data preprocessing process is tokenization. Tokenization is the process of breaking down text into small parts called tokens, usually single words. This process aims to separate each element in a sentence so that it can be explained individually by the modeling algorithm. In this study, tokenization was carried out on tweet text that had been cleaned and normalized previously. For example, the sentence "is not surprised by the presidential election results announced by the KPU tonight ..." will be changed into a series of words such as ['already', 'not', 'surprised', 'with', 'results', 'presidential election', 'which', 'di', 'announce', 'kpu', 'night', 'ini']. This process is very important because it allows each word to be identified as a feature that can be used for sentiment analysis. With tokenization, the model can understand the context of words in a sentence and separate words that have significant meaning. Tokenization is also a crucial initial stage before further processes such as removing stop words, stemming, and extracting features using the TF-IDF method are carried out.

	normalized_text	tokenized_text
1440	sudah tidak kaget dengan hasil pilpres yang di umumkan kpu malam ini sudah dari awal sejak mk dan kpu meloloskan gibran menjadi calon wakil prabowo menjadi salah satu kandidat capres pemilu bak sinetron yang kita sudah tau di mana ujung ceritanya mari kita tunggu sinetron episode berikut nya	['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini', 'sudah', 'dari', 'awal', 'sejak', 'mk', 'dan', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'yang', 'kita', 'sudah', 'tau', 'di', 'mana', 'ujung', 'ceritanya', 'mari', 'kita', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']
1441	pemilu hasil bansos 465 t	['pemilu', 'hasil', 'bansos', '465', 't']
1442	pemilu kok dianggap perang kacau anda ini ketum nasdem saja sebagai partai penyokong utama paslon no 1 sudah menyatakan menerima hasil pemilu mosok anda tidak cepat cuci muka sono biar siuman	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'anda', 'ini', 'ketum', 'nasdem', 'saja', 'sebagai', 'partai', 'penyokong', 'utama', 'paslon', 'no', '1', 'sudah', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'anda', 'tidak', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']
1443	tetap lawan sampai tolak pemilu curang tidak mengakui pemimpin dari hasil kecurangan yang tidak beradab	['tetap', 'lawan', 'sampai', 'tolak', 'pemilu', 'curang', 'tidak', 'mengakui', 'pemimpin', 'dari', 'hasil', 'kecurangan', 'yang', 'tidak', 'beradab']
1444	saya minta kebijaksanaan mui yang saat ini diketuai kyai mengenai hasil pemilu 2024 sebagai umat islam kita dituntut untuk adil cc	['saya', 'minta', 'kebijaksanaan', 'mui', 'yang', 'saat', 'ini', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', '2024', 'sebagai', 'umat', 'islam', 'kita', 'dituntut', 'untuk', 'adil', 'cc']
1446	kok ngumumin hasil pemilu yang amat penting tengah malem bangett dahh orangorang juga rata2 sudah pada tidur setelah tarwih	['kok', 'ngumumin', 'hasil', 'pemilu', 'yang', 'amat', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'juga', 'rata2', 'sudah', 'pada', 'tidur', 'setelah', 'tarwih']
1447	dari hasil pemilu kali ini kita mengetahui bahwa mayoritas penduduk indonesia 1 minim literasi 2 tidak suka perdebatan yang berdasarkan teori data dan fakta 3 cenderung duniawi 4 sekarang ya sekarang buat besok pikirinnya besok aja lagi 5 lebih suka kekerasan cont	['dari', 'hasil', 'pemilu', 'kali', 'ini', 'kita', 'mengetahui', 'bahwa', 'mayoritas', 'penduduk', 'indonesia', '1', 'minim', 'literasi', '2', 'tidak', 'suka', 'perdebatan', 'yang', 'berdasarkan', 'teori', 'data', 'dan', 'fakta', '3', 'cenderung', 'duniawi', '4', 'sekarang', 'ya', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lagi', '5', 'lebih', 'suka', 'kekerasan', 'cont']

Fig 4. Tokenization Result

After going through the tokenization stage, the next process in data preprocessing is stopword removal, which is the removal of words that are considered not to have a significant contribution to the meaning of the text. Stopwords are common words such as "yang", "dan", "di", "ini", "dari", and so on, which often appear in the text but do not provide important information in the context of sentiment analysis. In this study, the stopword removal process was carried out using the Sastrawi library, which provides a list of common words in Indonesian that are classified as stopwords. Each tokenized token will be checked and compared with the list, then removed if found in the list. For example, a tokenized sentence such as ['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini'] after being processed becomes ['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam'], with words such as "sudah", "yang", "di", and "ini" having been removed. This process helps reduce noise in the data and ensures that only important words are used in the next stages of analysis, such as stemming and feature extraction. Thus, stopword removal plays a vital role in improving the efficiency and accuracy of sentiment analysis models.

DOI: doi.org/10.35970/jinita.vxixx.xx

	tokenized_text	text_after_stopword
1440	['sudah', 'tidak', 'kaget', 'dengan', 'hasil', 'pilpres', 'yang', 'di', 'umumkan', 'kpu', 'malam', 'ini', 'sudah', 'dari', 'awal', 'sejak', 'mk', 'dan', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'yang', 'kita', 'sudah', 'tau', 'di', 'mana', 'ujung', 'ceritanya', 'mari', 'kita', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']	['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam', 'awal', 'sejak', 'mk', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'tau', 'mana', 'ujung', 'ceritanya', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']
1441	['pemilu', 'hasil', 'bansos', 't']	['pemilu', 'hasil', 'bansos', 't']
1442	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'anda', 'ini', 'ketum', 'nasdem', 'saja', 'sebagai', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'sudah', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'anda', 'tidak', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'ketum', 'nasdem', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']
1443	['tetap', 'lawan', 'sampai', 'tolak', 'pemilu', 'curang', 'tidak', 'mengakui', 'pemimpin', 'dari', 'hasil', 'kecurangan', 'yang', 'tidak', 'beradab']	['tetap', 'lawan', 'tolak', 'pemilu', 'curang', 'mengakui', 'pemimpin', 'hasil', 'kecurangan', 'beradab']

Fig 5. Stopword removal Result

The final process in the data preprocessing stage is stemming, which is the process of changing words that have affixes such as prefixes, suffixes, or a combination of both into a basic form (root word). The purpose of this process is to weave variations in word forms that have the same meaning, so as to improve data consistency and analysis effectiveness. In this study, the stemming process was carried out using the Sastrawi library, which is specifically designed to handle Indonesian language morphology. For example, words such as "ngumumin" are changed to "umum", "orangorang" to "orang", and "bangett" to "banget". After stemming, the form of tokens that have been combined will be recombined into plain text, which will be used in the next stage, namely feature extraction. With stemming, the number of word variations in the dataset can be minimized, so that the machine learning model can recognize patterns more accurately and efficiently. This process is very important, especially in handling unstructured data such as tweets, which contain many non-standard words and spelling variations.

	text_after_stopword	text_after_stemming
1440	['kaget', 'hasil', 'pilpres', 'umumkan', 'kpu', 'malam', 'awal', 'sejak', 'mk', 'kpu', 'meloloskan', 'gibran', 'menjadi', 'calon', 'wakil', 'prabowo', 'menjadi', 'salah', 'satu', 'kandidat', 'capres', 'pemilu', 'bak', 'sinetron', 'tau', 'mana', 'ujung', 'ceritanya', 'tunggu', 'sinetron', 'episode', 'berikut', 'nya']	kaget hasil pilpres umum kpu malam awal sejak mk kpu lolos gibran jadi calon wakil prabowo jadi salah satu kandidat capres pemilu bak sinetron tau mana ujung cerita tunggu sinetron episode ikut nya
1441	['pemilu', 'hasil', 'bansos', 't']	pemilu hasil bansos t
1442	['pemilu', 'kok', 'dianggap', 'perang', 'kacau', 'ketum', 'nasdem', 'partai', 'penyokong', 'utama', 'paslon', 'no', 'menyatakan', 'menerima', 'hasil', 'pemilu', 'mosok', 'cepat', 'cuci', 'muka', 'sono', 'biar', 'siuman']	pemilu kok anggap perang kacau tum nasdem partai sokong utama paslon no nyata terima hasil pemilu mosok cepat cuci muka sono biar siuman
1443	['tetap', 'lawan', 'tolak', 'pemilu', 'curang', 'mengakui', 'pemimpin', 'hasil', 'kecurangan', 'beradab']	tetap lawan tolak pemilu curang aku pimpin hasil curang adab
1444	['minta', 'kebijaksanaan', 'mui', 'diketuai', 'kyai', 'mengenai', 'hasil', 'pemilu', 'umat', 'islam', 'dituntut', 'adil', 'cc']	minta bijaksana mui tuai kyai kena hasil pemilu umat islam tuntut adil cc
1446	['kok', 'ngumumin', 'hasil', 'pemilu', 'penting', 'tengah', 'malem', 'bangett', 'dahh', 'orangorang', 'rata', 'tidur', 'tarwih']	kok ngumumin hasil pemilu penting tengah malem bangett dahh orangorang rata tidur tarwih
1447	['hasil', 'pemilu', 'kali', 'mengetahui', 'mayoritas', 'penduduk', 'indonesia', 'minim', 'literasi', 'suka', 'perdebatan', 'berdasarkan', 'teori', 'data', 'fakta', 'cenderung', 'duniawi', 'sekarang', 'sekarang', 'buat', 'besok', 'pikirinnya', 'besok', 'aja', 'lebih', 'suka', 'kekerasan', 'cont']	hasil pemilu kali tahu mayoritas duduk indonesia minim literasi suka debat dasar teori data fakta cenderung duniawi sekarang sekarang buat besok pikirinnya besok aja lebih suka keras cont

Fig 6. Stemming Result

C. LEXICON BASED LABELLING

At this stage, labeling of text data that has been previously processed will be carried out using a lexicon-based approach. At this stage, a dictionary has been prepared containing the words positive sentiment and also negative sentiment. In this section, it is necessary to mention the image used as a reference in the explanatory sentence. After labeling is carried out, the data distribution for each positive, negative, and neutral sentiment is shown in the reference Fig. 7.

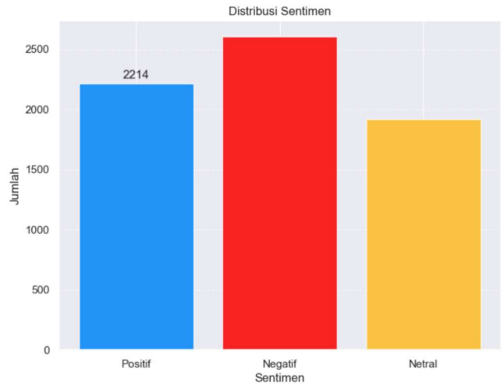


Fig 7. Sentiment Distribution

D. FEATURE EXTRACTION TF-IDF

In this process, feature extraction is carried out using the Term Frequency-Inverse Document Frequency (TF-IDF) method to convert text data into a numerical format that can be processed by a machine learning model. The results of the TF-IDF process produce 7256 features or words which have their respective weights in vector form. Figure 4.12 is an example of TF-IDF features and their weights in each document. The columns in the table represent each word in the entire sentence, while each row represents the sequence of the document or text. This arrangement is illustrated in the reference image (see Figure 8).

	bansos	curang	indonesia	mahkamah	pemilu	pilpres	prabowo	tolak
2993	0.000000	0.278981	0.169016	0.000000	0.126922	0.000000	0.000000	0.000000
2994	0.000000	0.000000	0.000000	0.168127	0.000000	0.121763	0.000000	0.000000
2995	0.000000	0.000000	0.000000	0.205216	0.000000	0.074312	0.000000	0.000000
2996	0.000000	0.000000	0.000000	0.000000	0.000000	0.124070	0.196398	0.000000
2997	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.165893	0.000000
2998	0.000000	0.000000	0.186157	0.126858	0.000000	0.091874	0.000000	0.000000
2999	0.000000	0.000000	0.000000	0.000000	0.109154	0.000000	0.227116	0.000000

Fig 8. Word in the entire sentence

E. STACKING MODELLING

At this stage, a sentiment analysis model will be built using the ensemble learning stacking method involving three algorithms as the base model, namely Naive Bayes, Random Forest, and Support Vector Machine, as well as Random Forest as a meta model. The data used in creating this model is divided in a ratio of 80:20, where 80% of the data is used for training and 20% for testing. The result is 5389 training data and 1348 testing data. The following represents sample data, as depicted in the reference image (see Figure 9).

	text_after_stemming	sentiment
150	hiv indonesia raya tingkat azab praroro nepo baby pemenang pemilu	Negatif
151	terus lampias marah presiden kalah pilpres hasil resmi kpu rubah apa presiden wakil presiden pilih sah	Netral
152	paman mahkamah konstitusi mbatalin hasil pemilu wakanda wakanda	Negatif
153	tidak banjir pesisir utara minggu minggu ganjar duluan terjun asih bantu presiden capres pemenang pemilu tidak nilai sombong rakyat	Negatif
154	silaturahmi maksud politisasi terima hasil pemilu	Positif
155	kuat argumen mahkamah konstitusi tidak rubah hasil pilpres gandeng pilpres politikus malin kundang anak mantu cundang	Netral
156	temu tidak hak tuan rumah anies tolak hak mas gibran temu anies sosok anies diri diri sengketa hasil pilpres belum selesai	Negatif
157	kpu umum pemenang pilpres pasang prabowo gibran menang jokowi versi secc akun	Positif
158	ambil contoh pp muhammadiyah sikap hasil pemilu dewasa sabar	Positif
159	bangga juara pileg tidak terima hasil pilpres otak kerdil	Netral

Fig 9. Training Data Sample

	text_after_stemming	sentiment
150	agam hasil pemilu cermin agam masyarakat hargai	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	Negatif
152	mahkamah konstitusi pas rapat rekapitulasi manis tidak tau umum pilpres tau mahkamah konstitusi timses ngumpulin bukti tidak salah pakai advokat kurang	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	Positif
154	malam ngobrolin hasil pemilu pajak pacar gara rekonsiliasi israel	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	Positif
156	surya paloh terima hasil pemilu sahabat presiden	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	Positif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	Negatif
159	tiktok ribut hasil pemilu kali ribut hasil pemilu ribut kpop sosmed irl ngantuk kerja stngh	Netral

Fig 10. Testing Data Sample

After the data is shared, each base model will be trained using the training data and will later produce predictions on the test data. Through the predict_proba function, each model will provide a probability for whether the data is labeled positive, negative or neutral. The class or label that has the highest probability will be used as the final prediction of the model in the Figure 11.

Commented [5]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

```
# Membuat meta-feature untuk training data
nb_train_pred = nb.predict_proba(X_train_tfidf)
rf_train_pred = rf.predict_proba(X_train_tfidf)
svm_train_pred = svm.predict_proba(X_train_tfidf)
meta_features_train = np.concatenate([nb_train_pred, rf_train_pred, svm_train_pred], axis=1)

# Membuat meta-feature untuk test data
nb_test_pred_proba = nb.predict_proba(X_test_tfidf)
rf_test_pred_proba = rf.predict_proba(X_test_tfidf)
svm_test_pred_proba = svm.predict_proba(X_test_tfidf)
meta_features_test = np.concatenate([nb_test_pred_proba, rf_test_pred_proba, svm_test_pred_proba], axis=1)

# Inisialisasi dan training meta learner
meta_learner = RandomForestClassifier(n_estimators=100, random_state=42)
meta_learner.fit(meta_features_train, y_train)

# Prediksi akhir dengan meta learner pada test data
final_predictions = meta_learner.predict(meta_features_test)
```

Fig 11. Stacking Code

Figure 2 shows the prediction results and the class probability of each model against the test data. The order of classes 0, 1 and 2 in the table shows the negative, neutral and positive classes in the Figure 12.

Commented [6]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

	text	nb_class0	nb_class1	nb_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.016227	0.027230	0.956543	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.622731	0.188593	0.188675	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.348763	0.402374	0.248863	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.360907	0.252019	0.387073	Positif
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.185071	0.594748	0.220181	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.300622	0.151513	0.547865	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.015130	0.032929	0.951941	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.564754	0.209396	0.225850	Negatif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.620861	0.141082	0.238057	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.427569	0.341746	0.230685	Negatif

Fig 12. Naïve Bayes Probability Result

	text	rf_class0	rf_class1	rf_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.081357	0.262206	0.656437	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.548884	0.226526	0.224590	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.129437	0.766280	0.104283	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.102069	0.206751	0.691180	Positif
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.455056	0.416384	0.128561	Negatif
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.277810	0.125952	0.596238	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.090937	0.169992	0.739071	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.314492	0.410989	0.274519	Netral
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.411056	0.261349	0.327595	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.675032	0.170857	0.154111	Negatif

Fig 13. Random Forest Probability Result

	text	svm_class0	svm_class1	svm_class2	predict
150	agam hasil pemilu cermin agam masyarakat hargai	0.015999	0.139643	0.844358	Positif
151	maaf tidak sudi presiden libat culi wakil hasil begal keputusan mahkamah konstitusi tidak terima hasil pilpres hasil curang jokowi teman tenannya	0.742923	0.206424	0.050653	Negatif
152	pdip pemenang pemilu jokowi google pdip partai jokowi jokowi jokowi partai	0.134347	0.786524	0.079129	Netral
153	aju mohon kait selisih hasil pemilihan phpu pilpres prabowo subianto	0.023798	0.508969	0.467233	Netral
154	hasil persentasi pemilu presiden abang maksud dpr dprd tidak anies gerindra bukti anies tidak pengaruh	0.138457	0.789167	0.072376	Netral
155	menang mutlak kena prabowo gibran legowo jangan pakai pengadilan rakyat buntut runyam pendukung prabowo gibran tidak rela hasil pemilu presiden batal kpu orang pintar pintar profesional	0.002719	0.012429	0.984851	Positif
156	surya paloh terima hasil pemilu sahabat presiden	0.000480	0.015403	0.984117	Positif
157	benar orang tuntut benar rebut benar rebut jujur benar ukur patok suara mayoritas hasil pemilu dasar sejarah angka angka tipu tuntut jujur benar	0.411272	0.226814	0.361914	Negatif
158	rakyat usaha kena pajak bayar pajak butuh pasti hukum pemenang pilpres moga pasti mukidi oktober lengser negara milik rakyat milik pribadi bibik	0.515261	0.302810	0.181929	Negatif
159	ketawa sudut kubu gimmik tutup kurang kursi tunggu sidang hasil gugat demokrat mahkamah konstitusi demokrat tidak mahkamah konstitusi pemilu xixixixi	0.461814	0.378116	0.160070	Negatif

Fig 14. SVM Probability Result

After all base models have their own predictions, the prediction results will be combined and used as features of the meta model. So the meta model will carry out training and testing data using these new features. The following is an example of a feature that will be used by the meta model to make final predictions in figure 15.

Commented [7]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

DOI: doi.org/10.35970/jinita.vxixx.xx

	nb_class0	nb_class1	nb_class2	rf_class0	rf_class1	rf_class2	svm_class0	svm_class1	svm_class2	predict
150	0.016227	0.027230	0.956543	0.081357	0.262206	0.656437	0.015628	0.135144	0.849228	Positif
151	0.622731	0.188593	0.188675	0.548884	0.226526	0.224590	0.742093	0.207783	0.050124	Negatif
152	0.348763	0.402374	0.248863	0.129437	0.766280	0.104283	0.137193	0.784752	0.078055	Netral
153	0.360907	0.252019	0.387073	0.102069	0.206751	0.691180	0.023476	0.507070	0.469453	Positif
154	0.185071	0.594748	0.220181	0.455056	0.416384	0.128561	0.141315	0.787167	0.071518	Negatif
155	0.300622	0.151513	0.547865	0.277810	0.125952	0.596238	0.002552	0.011500	0.985948	Positif
156	0.015130	0.032929	0.951941	0.090937	0.169992	0.739071	0.000436	0.014297	0.985267	Positif
157	0.564754	0.209396	0.225850	0.314492	0.410989	0.274519	0.410736	0.224927	0.364337	Netral
158	0.620861	0.141082	0.238057	0.411056	0.261349	0.327595	0.515909	0.302254	0.181837	Positif
159	0.427569	0.341746	0.230685	0.675032	0.170857	0.154111	0.462122	0.378279	0.159599	Negatif

Fig 15. Stacking Dataset And Result

F. EVALUATION

At this stage, all models that have been trained and have produced predictions will be evaluated to see their performance using the Confusion Matrix. The evaluation matrix used includes accuracy, precision recall, and F1-score. Each matrix will be calculated using the following formula

The Confusion Matrix for the Naïve Bayes model shows that the model succeeded in predicting correctly (True Positive) 323 data with positive sentiment, 105 data with neutral sentiment, 473 data with negative sentiment in figure 16.

Commented [8]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

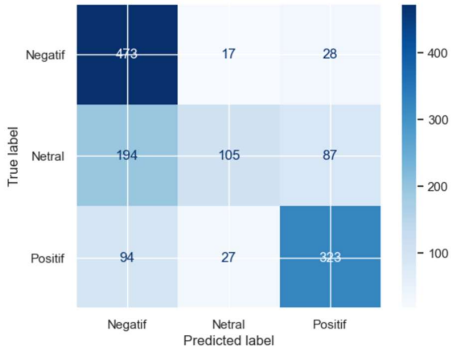


Fig 16. Naïve bayes Model's Confusion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the Naïve Bayes model

$$Accuracy = \frac{323 + 105 + 403}{1348} = \frac{829}{1348} = 0.6684$$

The accuracy of the Naïve Bayes model is 0.6684

$$Precision\ Positif = \frac{323}{323 + 115} = \frac{323}{438} = 0.7374$$

$$Precision\ Netral = \frac{105}{105 + 44} = \frac{105}{149} = 0{,}7047$$

$$Precision\ Negatif = \frac{473}{473 + 288} = \frac{473}{761} = 0{,}6216$$

$$Precision_{weighted} = \frac{(444 \times 0{,}7347) + (386 \times 0{,}7047) + (518 \times 0{,}6216)}{444 + 386 + 518} = 0.6835$$

The precision of the Naïve Bayes model is 0.6835

$$Recall\ Positif = \frac{323}{323 + 121} = \frac{323}{444} = 0{,}7275$$

$$Recall\ Netral = \frac{105}{105 + 281} = \frac{105}{386} = 0.272$$

$$Recall\ Negatif = \frac{473}{473+45} = \frac{473}{518} = 0.9131$$

$$Recall_{weighted} = \frac{(444 \times 0{,}7275) + (386 \times 0.272) + (518 \times 0.9131)}{444+386+518} = 0,6684$$

The recall of the Naïve Bayes model is 0.6684

$$F-1\ Score\ Positif = 2\ x\ \frac{(0{,}7374 \times 0{,}7275)}{(0{,}7374+0{,}7275)} = 0{,}7324$$

$$F-1\ Score\ Netral = 2\ x\ \frac{(0{,}7047 \times 0{,}272)}{(0{,}7047+0{,}272)} = 0{,}3925$$

$$F-1\ Score\ Negatif = 2\ x\ \frac{(0{,}6216 \times 0{,}9131)}{(0{,}6216+0{,}9131)} = 0{,}7397$$

$$F-1Score_{weighted} = \frac{(444 \times 0{,}7324) + (386 \times 0.3925) + (518 \times 0.7397)}{444+386+518} = 0,6379$$

The F-1 Score of the Naïve Bayes model is 0.6379

The Confusion Matrix for the Random Forest model shows that the model succeeded in predicting correctly (True Positive) 348 data with positive sentiment, 232 data with neutral sentiment, 428 data with negative sentiment in Figure 17.

Commented [9]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

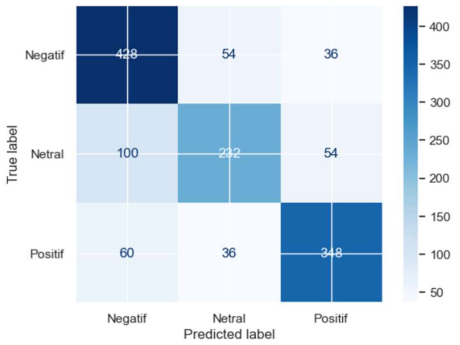


Fig 17. Random Forest Model's Confussion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the Random Forest model

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{348+232+428}{1348} = \frac{1008}{1348} = 0.7478$$

The accuracy of the Random Forest model is 0.7478

$$Precision\ Positif = \frac{348}{348+90} = \frac{348}{438} = 0.7945$$

$$Precision\ Netral = \frac{232}{232+90} = \frac{232}{222} = 0{,}7205$$

$$Precision\ Negatif = \frac{428}{428+160} = \frac{428}{588} = 0{,}7279$$

$$Precision_{weighted} = \frac{(444 \times 0{,}7945)\ + (386 \times 0{,}7205) + (518 \times 0{,}7279)}{444+386+518} = 0.7477$$

The precision of the Random Forest model is 0.7477

$$Recall\ Positif = \frac{348}{348+96} = \frac{348}{444} = 0{,}7838$$

$$Recall\ Netral = \frac{232}{232+154} = \frac{232}{386} = 0.601$$

$$Recall\ Negatif = \frac{428}{428+90} = \frac{428}{518} = 0.8263$$

$$Recall_{weighted} = \frac{(444 \times 0{,}7838) + (386 \times 0.601) + (518 \times 0.8263)}{444+386+518} = 0,7478$$

The recall of the Random Forest model is 0.7478

$$F-1\ Score\ Positif = 2\ x\ \frac{(0{,}7945 \times 0{,}7838)}{(0{,}7945+0{,}7838)} = 0{,}7891$$

$$F-1\ Score\ Netral = 2\ x\ \frac{(0{,}7205 \times 0{,}601)}{(0{,}7205+0{,}601)} = 0{,}6553$$

$$F-1\ Score\ Negatif = 2\ x\ \frac{(0{,}7279 \times 0{,}8263)}{(0{,}7279+0{,}8263)} = 0{,}774$$

$$F-1Score_{weighted} = \frac{(444 \times 0{,}6718) + (386 \times 0.5771) + (518 \times 0.7157)}{444+386+518} = 0,745$$

The F-1 Score of the Random Forest model is 0.745

The Confusion Matrix for the SVM model shows that the model succeeded in predicting correctly (True Positive) 370 data with positive sentiment, 252 data with neutral sentiment, 426 data with negative sentiment in Figure 18.

Commented [10]: In this section, it is necessary to mention the image used as a reference in the explanatory sentence.

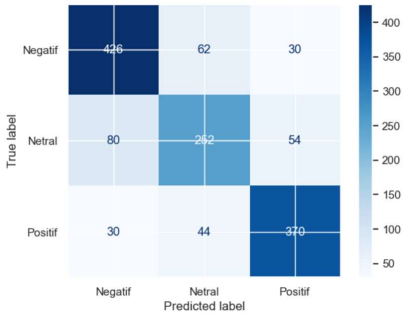


Fig 18. SVM Model's Confusion Matrix

Below are calculations to find the accuracy, precision, recall, and F1-score of the SVM model

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{370+252+426}{13512} = \frac{1048}{1351} = 0.7774$$

The accuracy of the SVM model is 0.7774

$$Precission\ Positif = \frac{370}{370+84} = \frac{370}{454} = 0.815$$

$$Precission\ Netral = \frac{252}{252+106} = \frac{252}{358} = 0{,}7039$$

$$Precission\ Negatif = \frac{426}{426+110} = \frac{426}{536} = 0{,}7984$$

$$Precission_{weighted} = \frac{(444 \times 0{,}815)\ + (386 \times 0{,}7039) + (518 \times 0{,}7984)}{444+386+518} = 0.7754$$

The precission of the naïve Bayes model is 0.7754

$$Recall\ Positif = \frac{370}{370+74} = \frac{370}{444} = 0{,}8333$$

$$Recall\ Netral = \frac{252}{252+134} = \frac{252}{386} = 0.6528$$

$$Recall\ Negatif = \frac{426}{426+92} = \frac{426}{518} = 0.8224$$

$$Recall_{weighted} = \frac{(444 \times 0{,}8333) + (386 \times 0.6528) + (518 \times 0.8224)}{444+386+518} = 0,7774$$

The recall of the SVM model is 0.7774

$$F-1\ Score\ Positif = 2\,x\,\frac{(0{,}815 \times 0{,}8333)}{(0{,}815+0{,}8333)} = 0{,}824$$

$$F-1\ Score\ Netral = 2\,x\,\frac{(0{,}7039 \times 0{,}6528)}{(0{,}7039+0{,}6528)} = 0{,}6774$$

$$F-1\ Score\ Negatif = 2\,x\,\frac{(0{,}7984 \times 0{,}8224)}{(0{,}7984+0{,}8224)} = 0{,}8084$$

$$F-1Score_{weighted} = \frac{(444 \times 0{,}824) + (386 \times 0.6774) + (518 \times 0.8084)}{444+386+518} = 0,8084$$

The F-1 Score of the SVM model is 0.8084

The Confusion Matrix for the stacking model shows that the model succeeded in predicting correctly (True Positive) 396 data with positive sentiment, 256 data with neutral sentiment, 447 data with negative sentiment in Figure 19.

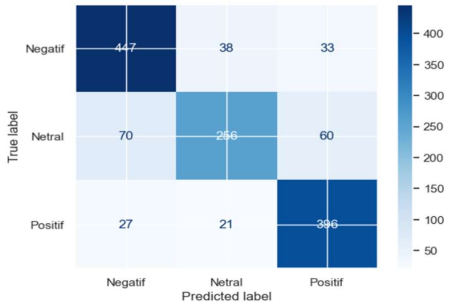


Fig 19. Stacking Model's Confusion Matrix

$$Accuracy = \frac{Total\ True\ Positives\ (TP)}{Total\ Sample} = \frac{396+256+447}{13512} = \frac{1099}{1351} = 0.8153$$

Below are calculations to find the accuracy, precision, recall, and F1-score of the Stacking model

$$Precision\ Positif = \frac{396}{396+93} = \frac{396}{489} = 0.8089$$

$$Precision\ Netral = \frac{256}{256+59} = \frac{256}{315} = 0.8127$$

$$Precision\ Negatif = \frac{447}{447+97} = \frac{447}{544} = 0{,}8127$$

$$Precision_{weighted} = \frac{(444 \times 0{,}8089)\ + (386 \times 0{,}8127) + (518 \times 0{,}8127)}{444+386+518} = 0.8152$$

The accuracy of the Stacking model is 0.8152

$$Recall\ Positif = \frac{396}{396+48} = \frac{396}{444} = 0{,}8919$$

$$Recall\ Netral = \frac{256}{256+130} = \frac{256}{386} = 0.6632$$

$$Recall\ Negatif = \frac{447}{447+71} = \frac{447}{518} = 0.8629$$

$$Recall_{weighted} = \frac{(444 \times 0{,}8919) + (386 \times 0.6632) + (518 \times 0.8629)}{444+386+518} = 0,8153$$

The recall of the Stacking model is 0.8153

$$F-1\ Score\ Positif\ =\ 2\ x\ \frac{(0{,}8098 \times 0{,}8919)}{(0{,}8098 + 0{,}8919)} = 0{,}8489$$

$$F-1\ Score\ Netral\ =\ 2\ x\ \frac{(0{,}8127 \times 0{,}6632)}{(0{,}8127 + 0{,}6632)} = 0{,}7304$$

$$F-1\ Score\ Negatif\ =\ 2\ x\ \frac{(0{,}8217 \times 0{,}8629)}{(0{,}8217 + 0{,}8629)} = 0{,}8418$$

$$F-1 Score_{weighte}\ =\ \frac{(444 \times 0{,}6718) + (386 \times 0.5771) + (518 \times 0.7157)}{444 + 386 + 518} = 0,8122$$

The F-1 Score of the Stacking model is 0.8122

The Result of all model can be seen in the table below

Table 3. Results Of Model

Commented [11]: In the table format section, it needs to be adjusted to the provisions in the JINITA journal.

Model	Accuracy	Precision	Recall	F-1 Score
Naïve Bayes	0.6684	0.6835	0,6684	0,6379
Support Vector Machine	0.7774	0.7754	0,7774	0,776
Random Forest	0.7478	0.7477	0,7478	0,745
Ensemble Stacking (RF)	0.8153	0.8152	0,8153	0,8122

4. CONCLUSION

As a result of this experiment, an ensemble learning stacking model was formed with several different base models, namely the SVM, Random Forest and Naïve Bayes algorithms. Each model carries out training and predictions on sentiment analysis data. The results, starting from the lowest, are the Naïve Bayes algorithm with an accuracy of 66.84%, followed by Random Forest with an accuracy of 74.78%, and the highest is SVM with an accuracy of 77.74%. The results of the three base models are compiled and used as input for a meta model that uses the Random Forest algorithm. The results show that the stacking ensemble method applied produces better accuracy than a single classifier, namely 81.53%. The implementation of ensemble learning through stacking, combining SVM, Random Forest, and Naïve Bayes as base models with a Random Forest meta-model, significantly enhances the accuracy and robustness of sentiment analysis on unstructured text data, demonstrating its effectiveness as a key contribution of this research.. The findings in this study not only demonstrate the success of the stacking technique in improving the accuracy of sentiment analysis, but also have important applications in social and practical contexts. In practice, this model can be applied by government agencies, media, or research organizations to automatically aggregate public opinion on national issues, such as election results. This allows for more responsive and accurate data-driven decision-making. In addition, this study contributes to the development of a robust machine learning model for unstructured data in Indonesian, which has so far been limited in the literature. Further research can explore this integration model with deep learning or apply it in different domains such as consumer opinion or public services.

REFERENCES

[1] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, 2023, doi: 10.1016/j.jksuci.2023.01.014.

[2] F. Matloob *et al.*, "Software defect prediction using ensemble learning: A systematic literature review," *IEEE Access*, vol. 9, pp. 98754–98771, 2021, doi: 10.1109/ACCESS.2021.3095559.

[3] Y. Görmez, Y. E. Işık, M. Temiz, and Z. Aydın, "FBSEM: A Novel Feature-Based Stacked Ensemble Method for Sentiment Analysis," *Int. J. Inf. Technol. Comput. Sci.*, vol. 12, no. 6, pp. 11–22, 2020, doi: 10.5815/ijitcs.2020.06.02.

[4] P. Thiengburanathum and P. Charoenkwan, "SETAR: Stacking Ensemble Learning for Thai Sentiment Analysis Using RoBERTa and Hybrid Feature Representation," *IEEE Access*, vol. 11, no. August, pp. 92822–92837, 2023, doi: 10.1109/ACCESS.2023.3308951.

[5] J. Gu, S. Liu, Z. Zhou, S. R. Chalov, and Q. Zhuang, "A Stacking Ensemble Learning Model for Monthly Rainfall Prediction in the Taihu Basin, China," *Water (Switzerland)*, vol. 14, no. 3, 2022, doi: 10.3390/w14030492.

[6] S. A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, "Stacking Strong Ensembles of Classifiers," in *IFIP Advances in Information and Communication Technology*, 2019. doi: 10.1007/978-3-030-19823-7_46.

[7] J. Dou *et al.*, "Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan," *Landslides*, vol. 17, no. 3, 2020, doi: 10.1007/s10346-019-01286-5.

[8] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, no. September, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[9] M. B. Alfarazi, M. 'Ariful Furqon, and H. Soepandi, "Sentiment Analysis of User Reviews for the Sister For Student Application Using Gaussian Naive Bayes and N-Gram," *J. Innov. Inf. Technol. Appl.*, pp. 101–108, 2024.

[10] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artif. Intell. Rev.*, vol. 55, no. 7, 2022, doi: 10.1007/s10462-022-10144-1.

[11] A. M. Iddrisu, S. Mensah, F. Boafo, G. R. Yeluripati, and P. Kudjo, "A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector," *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 2, 2023, doi: 10.1016/j.jjimei.2023.100180.

[12] R. Michaela Denise Gonzales and C. A. Hargreaves, "How can we use artificial intelligence for stock recommendation and risk management? A proposed decision support system," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, 2022, doi: 10.1016/j.jjimei.2022.100130.

[13] A. Tripathi, T. Goswami, S. K. Trivedi, and R. D. Sharma, "A multi class random forest (MCRF) model for classification of small plant peptides," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100029, 2021, doi: 10.1016/j.jjimei.2021.100029.

[14] M. Y. Aldean, P. Paradise, and N. A. Setya Nugraha, "Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 di Twitter Menggunakan Metode Random Forest Classifier (Studi Kasus: Vaksin Sinovac)," *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 4, no. 2, pp. 64–72, 2022, doi: 10.20895/inista.v4i2.575.

[15] A. Syafrianto, "Perbandingan Algoritma Naïve Bayes dan Decision Tree Pada Sentimen Analisis," *Indones. J. Comput. Sci. Res.*, vol. 1, no. 2, pp. 1–15, 2022, doi: 10.59095/ijcsr.v1i2.11.

[16] R. Nuraeni, A. Sudiarjo, and R. Rizal, "Perbandingan Algoritma Naïve Bayes Classifier dan Algoritma Decision Tree untuk Analisa Sistem Klasifikasi Judul Skripsi," *Innov. Res. Informatics*, vol. 3, no. 1, pp. 26–31, 2021, doi: 10.37058/innovatics.v3i1.2976.

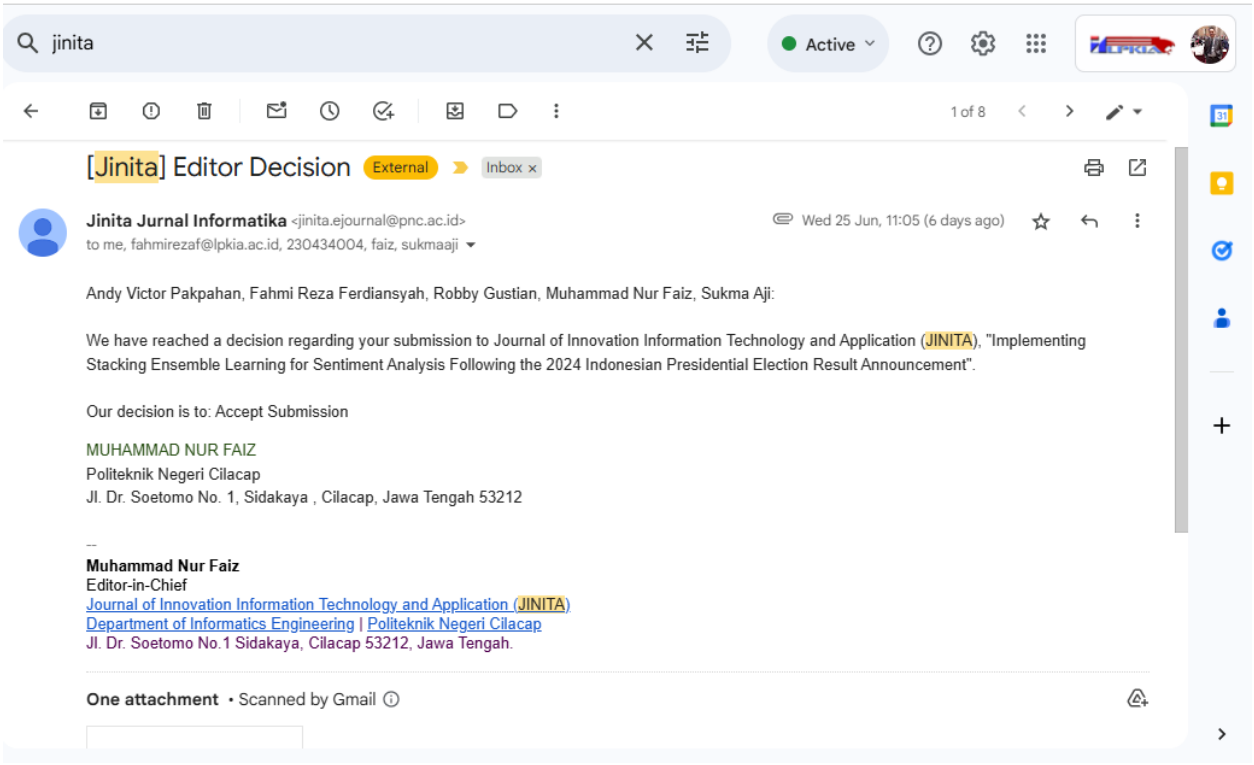
[17] S. H. Ramadhani and M. I. Wahyudin, "Analisis Sentimen Terhadap Vaksinasi Astra Zeneca pada Twitter Menggunakan Metode Naïve Bayes dan K-NN," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 4, 2022, doi: 10.35870/jtik.v6i4.530.

[18] M. Asgari, W. Yang, and M. Farnaghi, "Spatiotemporal data partitioning for distributed random forest algorithm: Air quality prediction using imbalanced big spatiotemporal data on spark distributed framework," *Environ. Technol. Innov.*, vol. 27, p. 102776, 2022, doi: 10.1016/j.eti.2022.102776.

[19] G. Meena, K. K. Mohbey, and S. Kumar, "Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach," *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 1, p. 100174, 2023, doi: 10.1016/j.jjimei.2023.100174.

[20] K. L. Tan, C. P. Lee, and K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074550.

7. JINITA Editor Decision – Accepted Submission – Letter of Acceptance (LoA)



Journal of Innovation Information Technology and Application (JINITA)
Address: Pusat Penelitian dan Pengabdian Masyarakat (P3M, Politeknik Negeri Cilacap
Jln. Dokter Soetomo No.1, Karangcengis Sidakaya Cilacap, Jawa Tengah 53212Telepon: (0282) 533329
Email: jinita.ejournal@pnc.ac.id

Letter of Acceptance (LoA)

Dear. Mr / Mrs Author of the paper,
Andy Victor Pakpahan, Fahmi Reza Ferdiansyah, Robby Gustian, Muhammad Nur Faiz, Sukma Aji;

We have reached a decision regarding your submission to Journal of Innovation Information Technology and Application (JINITA), "**Sentiment Analysis Using Stacking Ensemble After the 2024 Indonesian Election Results**" to be published in **Vol 7 no 1 June 2025**

Our decision is to: **Accept Submission.**

There are several things to note, namely:

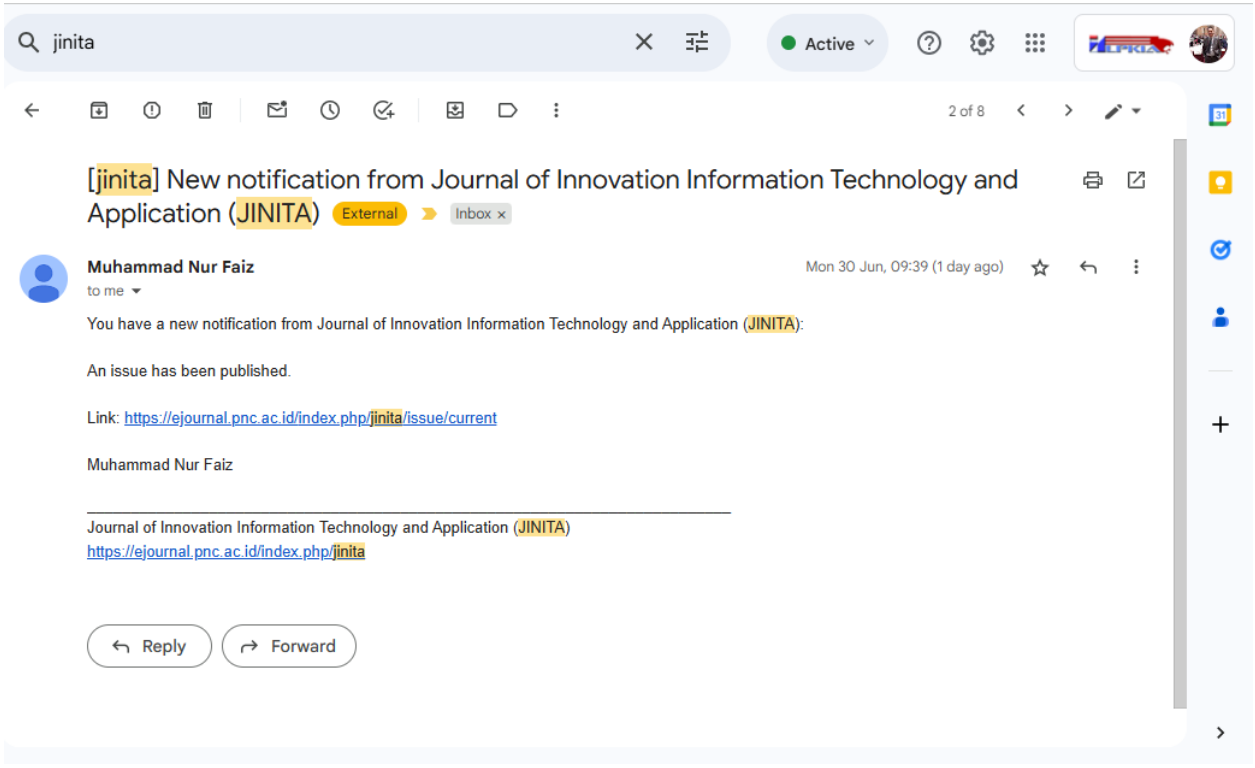
1. Please revise the OJS metadata and adjust the paper with the templates available on the Website. Paper must be written in INDONESIAN/ENGLISH. Abstract is made in *one languages (English and Indonesian)*. See templates and writing instructions on the website https://ejournal.pnc.ac.id/index.php/jinita.
2. Authors are required to fill out the COPYRIGHT FORM, completed and signed form is scanned and saved in PDF format.
3. The revised paper should be sent back to e-mail: jinita.ejournal@pnc.ac.id or OJS in the form of a .doc/.docx file. along with the COPYRIGHT FORM attachment that has been filled out and signed.
4. The author must pay attention to the metadata of the paper such as the author's name, affiliation, title, abstract and keywords in English,. Pay attention to the status paper on OJS.

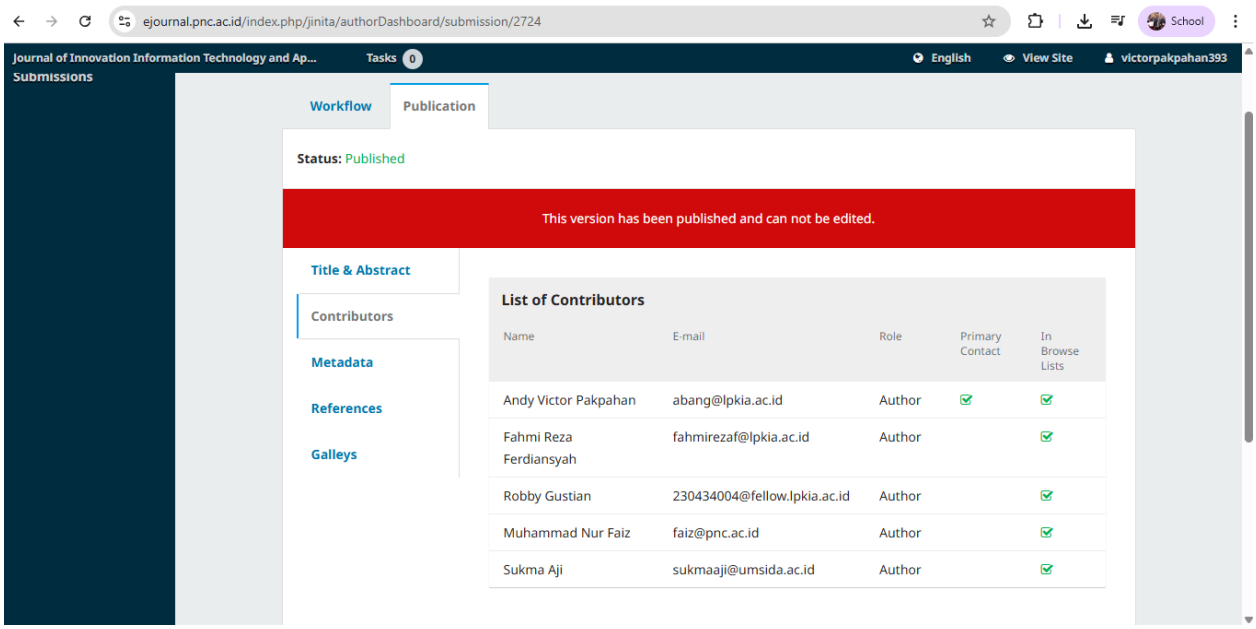
Cilacap, June 25th, 2025

Editor JINITA,
Editor-in-Chief,

Muhammad Nur Faiz, M.Kom

8. Publikasi Paper https://ejournal.pnc.ac.id/index.php/jinita/issue/current





Vol. 7 No. 1 (2025): JINITA, June 2025

This edition contains 16 articles, 170 pages, and 66 authors from 6 countries (Indonesia, D.R. Congo, Senegal, Benin, Iraq, Malaysia) which have been accepted and reviewed. The article was published in volume ,7 number 1 June 2025.

DOI: https://doi.org/10.35970/jinita.v7i1

Published: 2025-06-30

Full Issue

COVER, ED, BACK