J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

NLP Implementation For AI Generated Text Detection (ChatGPT) Using Naive Bayes Method

Rafel Fernando[1*], Yuliana Dewi Proboningrum[2], Septi Dwi Supriati[3], Nurmalitasari[4]

[1,2,3,4]*University of Duta Bangsa Surakarta, Faculty Computer Science, Information System, Jl. Bhayangkara No.55, Tipes, Serengan District, Surakarta City, Central Java 57154, Indonesia*

Keywords
NLP; Naive bayes Multinomial; Detection; Artificial Intelligence; ChatGPT

Corresponding Author:

220101031@mhs.udb.ac.id

Abstract
The rapid advancement of artificial intelligence (AI), particularly large language models such as Chat-GPT, has raised concerns regarding the authenticity and validity of digital content. The ability of AI to generate human-like text introduces risks of misuse, including plagiarism and information manipulation. This study aims to develop an AI-generated text detection system using the Multinomial Naive Bayes algorithm, which is widely used for text classification due to its simplicity and effectiveness. The Human Chat-GPT Comparison Corpus (H3C), sourced from Reddit's r/ELI5 subreddit, was used as the dataset and contains 800 question-and-answer entries generated by both humans and AI. During the labeling process, responses were consolidated into a single column and assigned labels based on their origin. Preprocessing steps included case folding, digit and punctuation removal, tokenization, stop-word removal, normalization, and finalization. Text features were extracted using the TF-IDF method, limited to the top 1,000 features. The model was trained on 80% of the data and tested on the remaining 20%. The results showed an accuracy of 93%, indicating that the Naive Bayes algorithm is effective in distinguishing AI-generated from human-generated text and holds promise as an automated tool for AI content detection

1. Introduction

Advances in digital technology have led to the emergence of machine learning and artificial intelligence (AI)[1]. One of the most significant developments is the large language model (LLM), which is capable of generating text with structure and writing style that closely resemble those of human language[2]. This advancement has had a major impact across various sectors, including education, journalism, and public services. However, it has also introduced a new challenge in the field of natural language processing (NLP): the authenticity of information in content generated by generative AI, such as Chat-GPT. NLP-based AI models are now able to produce text that is often indistinguishable from human writing[3]. This situation raises serious concerns regarding the validity of information and opens the door to potential misuse, including plagiarism, opinion

manipulation, and the spread of false information[5]. Therefore, there is a growing need for methods that can automatically detect and classify text in order to prevent misuse and maintain the integrity of information in the current digital era[4].

One of the most widely used large language models today is Chat-GPT, developed by OpenAI, which has the capability to automatically generate text with highly natural language structure. This presents challenges in terms of information validation, particularly in distinguishing between text written by humans and that generated by machines[5], [6]. Therefore, detecting AI-generated text is essential to ensure the authenticity of information and to prevent potential misuse of the technology. To address this issue, a system capable of detecting AI-generated content efficiently, accurately, and automatically is required. One commonly used approach in the field of Natural Language Processing (NLP) for text classification is the Naive Bayes method, which is known for its simplicity and effectiveness in various text-based classification tasks[7], [8].

The Naive Bayes algorithm is a simple yet effective classification method that is widely applied in text analysis tasks such as language identification and document categorization[9]. It operates by computing conditional probabilities based on the assumption of feature independence, which allows for efficient statistical modeling of linguistic data[10], [11]. Naive Bayes can identify patterns that are typical of machine-generated text and different from those found in human-written text[7]. When combined with appropriate pre-processing and training on relevant datasets, this method is expected to achieve high accuracy in distinguishing AI-generated from human-generated content. This contributes to efforts aimed at reducing the misuse of generative technologies in the digital environment.

According to Seldi and Leo, it was shown that the application of Multinomial Naive Bayes with the Bag of Words (BoW) approach to detect AI generated text produced an accuracy of 98%[12], [13], [14]. According to research by Muhammad Ziaul Haq et al., the application of classical Naive Bayes to detect hoaxes on social media managed to obtain an accuracy of 88% proving that the probabilistic approach remains relevant in dealing with unstructured data such as narrative text[4]. According to research by Guo et al. to distinguish human text and ChatGPT with the RoBERTa deep learning approach model, it can record an F1-score of 98%[2]. However, RoBERTa has limitations in terms of high computational requirements and implementation complexity[2]. According to Yadagiri et al [7], [15] detection using a trained deep learning model method combined with linguistic features produces 99.73% accuracy, but requires high performance with high computing resources and processing time. This provides a strong foundation for further exploration of Naive Bayes, especially the multinomial variant, as a lightweight and efficient approach to detecting text sources.

This study aims to implement the Naive Bayes method for detecting text generated by Chat-GPT. The method was selected due to its strong performance in text classification tasks and its computational efficiency. In this research, the Naive Bayes algorithm is applied in conjunction with several preprocessing steps in Natural Language Processing (NLP), including case folding, punctuation removal, tokenization, and stopword removal, to enhance data quality[5], [16]. The dataset consists of both human-generated and AI-generated text, sourced from the Reddit ELI5 platform. The findings of this study are expected to contribute to addressing the challenges of information authenticity in the current era of generative technology.

2. Research Method

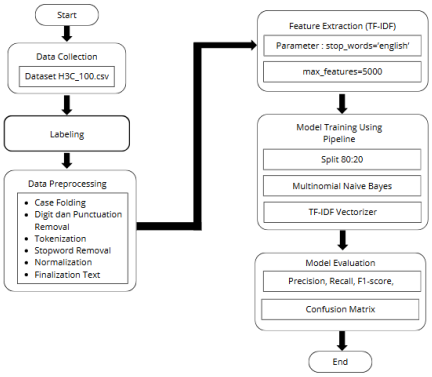

Figure 1. Research methodology

This study measures the level of accuracy of the naive Bayes algorithm model. The research method in Figure 1 can be seen in the sub-chapter below:

Data Collection

The data used in this article was obtained from GitHub https://github.com/Hello-SimpleAI/chatgpt-comparison-detection in the form of a collection of question and answer pairs from the Reddit platform, specifically from the ELI5 (Explain Like I'm 5) subreddit titled Human ChatGPT Comparison Corpus (H3C). The data length is 800 data and contains the columns id, question, human_answer, chatgpt_answer, and source.

Labeling

At this stage, the labeling process is carried out on the dataset consisting of a collection of texts generated by humans and by the ChatGPT model. This labeling process aims to provide an identity or category for each row of data that functions as training data for developing the classification model.

Data Preprocessing

- Case Folding is the text normalization stage by changing all characters in the text or sentence to lowercase[17], [18].
- Digit and Punctuation Removal is the stage of removing numbers and punctuation.
- Tokenization is the process of separating text into individual recognizable single word pieces (tokens).
- Stopword Removal is the removal of common words that often appear and are considered to have no significant meaning.
- Normalization is the process of changing non-standard words, abbreviations, or modified forms into standard forms that are in accordance with standard language[8]

Feature Extraction (TF-IDF)

TF-IDF is an effective feature extraction method in giving weight to each word based on its frequency of occurrence, so that it is able to represent the reks features numerically for classification purposes[10]. The parameters used are stop_words='english' in the TF-IDF calculation max_features=5000 which limits the number of unique features/words to 5000 relevant features. The effectiveness of TF-IDF in weighting textual data according to Roba et al. which states that the more frequently a word appears in a document, the higher the weight given[19]

Model Training Using Pipeline

- Multinomial Naive Bayes for text classification (human/ai) because Bayes decision theory is a fundamental statistical approach in pattern recognition. The standard form of Bayes' theorem is expressed as follows:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (1)$$

P (A|B) = Probability of A occurring given that B has occurred
P(A) = Probability of A occurring
P(B|A) = Probability of B occurring given that A has occurred
P(B) = Probability of B occurring

- This study uses a single data split method with 80% training data and 20% testing data, to evaluate model performance and ensure that overfitting does not occur.

Model Evaluation

- Precision Recall

$$\text{Precision} = 2\frac{TP}{TP+FP} \quad (2)$$

Description:
TP/True Positive : Positive class data classified as positive
TN/True Negative : Negative class data classified as negative
FP/False Positive : Positive class data classified as negative
FN/False Negative : Negative class data classified as positive

- Recall

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

- F1-Score

$$F1 = 2\frac{Precision*Recall}{Precision+Recall} \quad (4)$$

The model was also tested using a confusion matrix to determine the number of correct and incorrect predictions and to detect false positives/negatives.

3. Result and Discussions

Data Collection

In the data collection stage, the data contains a collection of question and answer pairs collected from the Reddit platform, specifically from the ELI5 (Explain Like I'm 5) subreddit named Human ChatGPT Comparison Corpus (H3C). The data length is 800 data containing the question, human_answer, chatgpt_answer, and source columns. The research data can be seen in Figure 2.

	question	human_answers	chatgpt_answers	source
0	Why is every book I hear about a " NY Times # ...	['Basically there are many categories of " Bes...	['There are many different best seller lists t...	reddit_eli5
1	If salt is so bad for cars , why do we use it ...	['salt is good for not dying in car crashes an...	["Salt is used on roads to help melt ice and s...	reddit_eli5
2	Why do we still have SD TV channels when HD lo...	["The way it works is that old TV stations got...	["There are a few reasons why we still have SD...	reddit_eli5
3	Why has nobody assassinated Kim Jong - un He i...	["You ca n't just go around assassinating the ...	['It is generally not acceptable or ethical to...	reddit_eli5
4	How was airplane technology able to advance so...	['Wanting to kill the shit out of Germans driv...	['After the Wright Brothers made the first pow...	reddit_eli5
...	...	...	...	...
795	Why do all talk shows always have the guest on...	["That 's moreso of an American thing . I like...	["There isn't any particular reason why talk s...	reddit_eli5
796	Why do humans have such a horrible sense of sm...	["Because we did nt need it . Our sense of sme...	['Human beings actually have a pretty good sen...	reddit_eli5
797	Medieval Nobility How did the system of nobili...	['> How did the system of nobility and peasant...	["The system of nobility and peasanthood, also...	reddit_eli5
798	What is the relationship between resolution (...	['The measurement you are looking for is dots ...	['The resolution of a photo, measured in pixel...	reddit_eli5
799	What is all the fuss about a Clockwork Orange ...	['A Clockwork Orange does n\'t have a " messag...	["A Clockwork Orange is a novel written by Ant...	reddit_eli5

Figure 2. Data collection

Labeling

In the labeling stage, the texts from human answers and ChatGPT are combined into one 'text' column and labeled 'human' or 'ai' separately, then combined into a DataFrame. The labeling results can be seen in Figure 3.

	text	label
1447	['Lobotomy is a surgical procedure in which th...	ai
1172	['Quarantine is a feature in antivirus softwar...	ai
1355	['Milk is more expensive than gas for a number...	ai
662	['News follows public interest . If people are...	human
1465	["A panic attack is a sudden and intense feeli...	ai

Figure 3. Labeling

Data Preprocessing

At this stage, several steps are carried out consisting of case folding, digit and punctuation removal, tokenization, stopword removal, and normalization.

- Case Folding

	text	casefolding
904	['In science, a theory is a well-supported exp...	['in science, a theory is a well-supported exp...
304	["You are experiencing radiative heating when ...	["you are experiencing radiative heating when ...
922	['Bugs like fruit flies are attracted to sweet...	['bugs like fruit flies are attracted to sweet...
423	['It depends on the size of the kitchen . In m...	['it depends on the size of the kitchen . in m...
1169	["Dry firing a bow is dangerous because it inv...	["dry firing a bow is dangerous because it inv...

Figure 4. Case folding

This process is done by applying the .str.lower() method to the 'text' column of the DataFrame by making all sentences lowercase. The casefolding stage in this study can be seen in Figure 4.

- Digit and Punctuation Removal

	casefolding	punctuation_removed
1276	["the celebration of christmas on december 25t...	the celebration of christmas on december th is...
494	['there are a few reasons , but the big ones a...	there are a few reasons but the big ones are ...
39	['filibustering is a tactic that evolved in th...	filibustering is a tactic that evolved in the ...
253	["because the website does n't hold the files ...	because the website does nt hold the files the...
1136	["the interview is a comedy film that was rele...	the interview is a comedy film that was releas...

Figure 5. Digit and Punctuation Removal

At this stage of the research, digit and punctuation removal was carried out on the text, where non-alphabetic characters (other than the letters 'a-z' and spaces) were removed from the 'casefolding' column. Digit and Punctuation Removal can be seen in Figure 5.

- Tokenization

	punctuation_removed	tokens
650	i would imagine that it s an instantaneous no...	[i, would, imagine, that, it, s, an, instantan...
1456	there is no law that requires employees to giv...	[there, is, no, law, that, requires, employees...
1064	in the context of the tv show the walking dead...	[in, the, context, of, the, tv, show, the, wal...
1101	orange juice is often kept in the refrigerated...	[orange, juice, is, often, kept, in, the, refr...
204	extrapolating from a few things i ve read in r...	[extrapolating, from, a, few, things, i, ve, r...

Figure 6. Tokenization

At this stage, the text tokenization process where sentences from the 'punctuation_removed' column are broken down into a list of individual words or 'tokens'. This process is implemented using the wordd_tokenize function from the nltk library. The tokenization stage in this study can be seen in Figure 6.

- Stopword Removal

	tokens	stopword_removed
17	[will, we, eventually, have, to, pay, it, off,...	[eventually, pay, nt, sorta, debt, household, ...
1553	[during, sleep, your, body, goes, into, a, sta...	[sleep, body, goes, state, muscle, paralysis, ...
229	[caution, hot, coffee, is, hot, caution, ice, ...	[caution, hot, coffee, hot, caution, ice, cold...
485	[feeling, depressed, as, in, sad, is, actually...	[feeling, depressed, sad, actually, quite, unl...
512	[think, about, all, of, the, things, that, cau...	[think, things, cause, car, accidents, drivers...

Figure 7. Stopword Removal

This stage involves stopword removal, eliminating frequently occurring words that carry little to no meaningful information and have been removed from the token list. This process is implemented by downloading a list of English stopwords from the NLTK library, then each token in the 'tokens' column is checked. The stopword removal stage can be seen in Figure 7.

- Normalization

	stopword_removed	normalized
516	[generally, legal, reasons, like, inheritance,...	[generally, legal, reasons, like, inheritance,...
1276	[celebration, christmas, december, th, based, ...	[celebration, christmas, december, th, based, ...
404	[could, probably, baby, years, ago, cognitivel...	[could, probably, baby, years, ago, cognitivel...
1249	[difficult, say, one, musician, better, anothe...	[difficult, say, one, musician, better, anothe...
768	[early, days, motoring, michelin, company, put...	[early, days, motoring, michelin, company, put...

Figure 8. Normalization

At the text normalization stage, where non-standard words or abbreviations in the 'stopword_removed' column are changed into standard form according to the specified normalization_dict. The Text Normalization Stage can be seen in Figure 8

- Finalization Text

	label	final_text
285	human	religions created time males dominant man crea...
532	human	equalizer device compensates less perfect devi...
1042	ai	iphone uses proprietary connector called light...
85	human	certain things happen certain times prior take...
971	ai	italy one major axis powers world war ii along...

Figure 9. Finalization Text

At the text finalization stage, where the list of normalized words from the 'normalized' column is recombined into one complete sentence string, and the results are stored in a new column named 'final_text'. The text finalization stage in this study can be seen in Figure 9.

TF-IDF Feature Extraction

The feature extraction process at this stage uses the TF-IDF method to represent text. Feature extraction is done using the TF-IDF Vectorizer library from Scikit-learn with the max_features=1000 parameter, which means that only the top 1000 words (based on frequency and IDF value) are selected as the main features. The selection of max features 1000 helps reduce overfitting and maintain computational efficiency without sacrificing model performance.

Model Training Using Pipeline

- Training Data and Testing Data

Figure 10. Training data and testing data

This stage involves dividing the data into two portions: 80% for training and 20% for testing. As visualized in Figure 10, AI data is marked in blue and human data in orange.

- Multinomial Naive Bayes

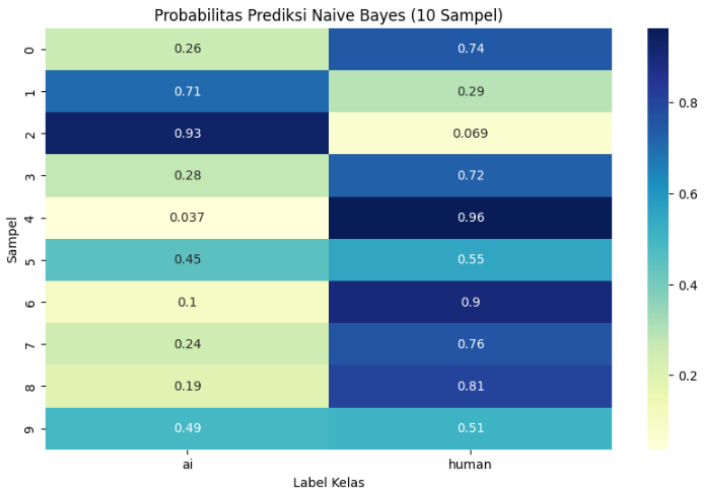

Figure 11. Naive Bayes Prediction

This step involves using the scikit-learn library to implement the Multinomial Naïve Bayes algorithm for prediction tasks. Testing was carried out on 10 samples, and the implementation results are presented in Figure 11.

- TF-IDF Vectorizer

Top 20 Fitur Berdasarkan TF-IDF (Training Set):

	Fitur	Total_TFIDF_Score
0	nt	53.492025
1	people	50.182078
2	may	38.175696
3	would	38.017306
4	like	35.804723
5	also	34.963628
6	one	31.394524
7	make	30.068168
8	way	29.289264
9	use	28.687480
10	different	28.637059
11	time	27.885324
12	might	26.306270
13	used	26.049133
14	important	25.212143
15	get	24.791595
16	could	24.321641
17	many	24.071558
18	water	22.585348
19	much	21.310263

Figure 12. TF-IDF Vectorizer

After training the model using the Multinomial Naïve Bayes algorithm, a new sample is tested to evaluate its prediction performance. During the feature extraction process with TF-IDF using the scikit-learn library, it was observed that there were 20 words or features with the highest TF-IDF scores. The results of the TF-IDF Vectorizer can be seen in Figure 12.

Model Evaluation

- Confusion Matrix

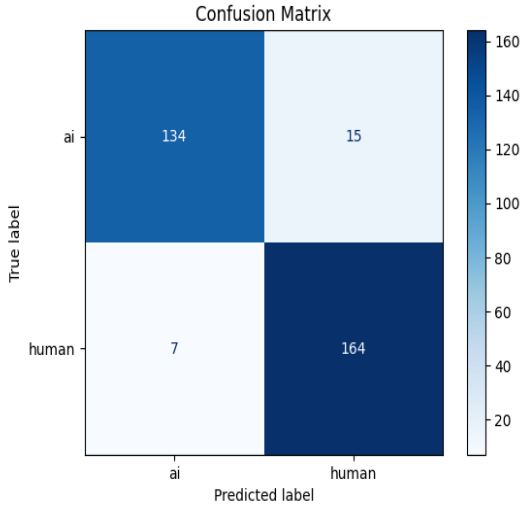

Figure 13. Confusion Matriks

The Naive Bayes model shows excellent performance in classifying AI and human texts. The number of false positive and false negative errors. Only 15 AI texts were misclassified as human texts, and only 7 human texts were misclassified as AI texts. This shows that the features of TF-IDF are quite effective in distinguishing the writing style characteristics of these two types of writers. The results of the confusion matrix implementation of the naive bayes method can be seen in Figure 13.

- Precision, Recall, F1-Score

	precision	recall	f1-score	support
ai	0.95	0.90	0.92	149
human	0.92	0.96	0.94	171
accuracy			0.93	320
macro avg	0.93	0.93	0.93	320
weighted avg	0.93	0.93	0.93	320

Figure 14. Evaluation

The evaluation metrics used to present the test results include precision, recall, and F1-score. The results indicate that the model achieved an accuracy of 93%, demonstrating its strong performance in accurately detecting and classifying text. The evaluation results in this study can be seen in Figure 14.

4. Conclusions and Future Works

This research successfully implemented and evaluated a Multinomial Naive Bayes classifier to distinguish between human-written and ChatGPT-generated text. By employing comprehensive NLP preprocessing and TF-IDF feature extraction on the Human ChatGPT Comparison Corpus (H3C) dataset , the model achieved a high accuracy of 93%. This finding demonstrates that a computationally efficient and interpretable algorithm can be highly effective for AI text detection, presenting a practical alternative to resource-intensive deep learning models. The results confirm that the Naive Bayes method provides excellent performance in automatically classifying text sources , offering a valuable contribution to addressing the challenges of information authenticity in the current technological landscape. To build upon this work, several avenues for future research are recommended. The model's robustness and generalizability could be tested on larger and more diverse datasets from various domains, such as news articles, academic writing, and different social

media platforms. Future studies could also explore more advanced feature engineering techniques, like word embeddings, to capture deeper semantic and stylistic nuances. Finally, a comparative analysis with other lightweight classification algorithms, such as Logistic Regression and Support Vector Machines, could be conducted to identify the most optimal model for this task. Integrating explainable AI (XAI) methods could also enhance model transparency by revealing which linguistic features are most indicative of AI-generated text

5. References

[1] L. Lim and S. Siripipatthanakul, "a Review of Artificial Intelligence (Ai) and Chatgpt Influencing the Digital Economy," no. December, pp. 2828–4925, 2023, doi: 10.47841/icorad.v2i2.139.

[2] & Y. W. Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, "How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection", doi: arXiv:2301.07597.

[3] Y. Su and Y. Wu, *Robust Detection of LLM-Generated Text: A Comparative Analysis*, vol. 1, no. 1. Association for Computing Machinery, 2024.

[4] D. J. Y. Y. G.-A. Odri, "Detecting generative artificial intelligence in scientific articles: Evasion techniques and implications for scientific integrity", doi: 10.1016/j.otsr.2023.103706.

[5] D. Biris, "Deep Learning Approaches for Detecting Text Generated by Artificial Intelligence," *Studia Universitatis Babeş-Bolyai Informatica*, vol. 69, no. 2, pp. 39–58, Apr. 2025, doi: 10.24193/subbi.2024.2.03.

[6] N. Islam, D. Sutradhar, H. Noor, J. T. Raya, M. T. Maisha, and D. M. Farid, "Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning," *arXiv preprint*, 2023, [Online]. Available: http://arxiv.org/abs/2306.01761

[7] A. Yadagiri, L. Shree, S. Parween, A. Raj, S. Maurya, and P. Pakray, "Detecting AI-Generated Text with Pre-Trained Models using Linguistic Features," *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pp. 188–196, 2024.

[8] D. O. Sihombing, "Implementasi NLP dan Cosine Similarity dalam Penilaian Ujian Esai Otomatis," *Jurnal Sistem Komputer dan Informasi*, vol. 4, no. 2, p. 396, 2022, doi: 10.30865/json.v4i2.5374.

[9] F. Novianti, K. Rizky, and N. Wardani, "Analisis Sentimen Masyarakat Terhadap Data Tweet Traveloka Menggunakan Naïve Bayes," *JIPI*, vol. 8, no. 3, pp. 922–993, 2023, doi: 10.29100/jipi.v8i3.3973.

[10] L. Azzahrah, "Naive Bayes Algorithm and TF-IDF for Detecting Plagiarism", doi: 10.33558/piksel.v12i2.9829.

[11] A. Shah, P. Ranka, U. Dedhia, and others, "Detecting and Unmasking AI-Generated Texts through Explainable Artificial Intelligence using Stylistic Features," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, pp. 1043–1053, 2023, doi: 10.14569/IJACSA.2023.01410110.

[12] L. S. Chanda, "Implementasi Algoritma Multinomial Naïve Bayes untuk Deteksi AI Generated Text".

[13] D. S. Kashid, J. D. Patil, and A. Buchade, "Live News Classification Using Naive Bayes Classifier," 2025, doi: 10.7759/s44389-024-01030-8.

[14] R. Rinaldi and R. Goejantoro, "Penerapan Metode Multinomial Naive Bayes: Studi Kasus PT Prudential Life Samarinda," *Eksponensial*, vol. 12, pp. 111–118, 2021, doi: 10.30872/eksponensial.v12i2.803.

[15] W. Widyawati and S. Sutanto, "Perbandingan Kinerja Naïve Bayes Multivariate dan Multinomial," *Journal of Innovative Future Technology*, vol. 2, no. 1, pp. 108–125, 2020, doi: 10.47080/iftech.v2i1.859.

[16] P. S. Putra, “Komentar Situs Reddit dengan Metode Lexicon Based,” 2024.

[17] N. Prova, “Detecting AI Generated Text Based on NLP and Machine Learning Approaches,” 2024.

[18] J. M. Polgan and others, “Algoritma Naïve Bayes untuk Mengidentifikasi Hoaks di Media Sosial,” *Jurnal Manajemen dan Profesi*, vol. 13, pp. 2020–2025, 2024, doi: 10.33395/jmp.v13i1.13937.

[19] A. S. R. Roba, S. Lailiyah, and A. Yusnita, “Application of Naive Bayes Algorithm for Analysis of User Reviews on Mobile Legends Game: Bang Bang,” *J-INTECH*, pp. 140–147, 2025, doi: 10.32664/j-intech.v13i01.1881.