Vol.3 No.2 October 2026: 63-76
International Journal of Computer Science and Humanitarian AI (IJCSHAI)
e-ISSN: 3064-4372
DOI: 110.21512/ijcshaijournal.v3i2.15283

Implementation of Fine-Tuned Model Llama3 on Hugging Face to detect Online Gambling Promotional Text

Muhammad Rafid Miftah Fadhil[1], Indira Syawanodya[2], Yulia Retnowati[3]

Software Engineering Study Program,
UPI Cibiru Campus,
West Java, Indonesia 40625
mrafid.mf@upi.edu; indira@upi.edu; yulia.retnowati@upi.edu

Abstract*—Online gambling is a serious problem in Indonesia due to its social and economic impact, worsened by ease of access and massive promotional strategies. From the past research, online gambling promotion in Indonesia has not been widely conducted. This study aims to implement the Llama 3.2 Large Language Model (LLM) based on the Hugging Face library to detect online gambling promotions and analyze its performance with and without optimization techniques. The Llama 3.2 model is fine-tuned using the Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA) approaches. Model performance was evaluated using the confusion matrix and classification report, covering metrics like accuracy, precision, recall, and F1-score. The results showed that the model without an optimizer produced an accuracy of 0.89. With the application of the AdamW optimizer, accuracy increased to 0.92 with an improvement in the balance between precision and recall in both classes. This improvement in performance demonstrates the effectiveness of using optimizers in improving model training and generalization processes. Overall, this study shows that Llama 3.2 is capable of effectively detecting online gambling promotions and can serve as a foundation for the development of illegal content detection systems.

***Kata Kunci*—Online Gambling; Llama 3.2; Hugging Face; Optimizer AdamW; Text Detection; PEFT; LoRA**

I. INTRODUCTION

Online gambling has become a serious socio-economic problem in Indonesia due to easy digitally access and massive promotional strategies, especially targeting teenagers and students. Based on the 2023 PPATK report, the value of online gambling transactions reached Rp327 trillion with more than 2.3 million active users, including nearly one million students. This activity poses risks of addiction, debt, crime, and mental health disorders.

Previous research has shown that the use of artificial intelligence (AI) can detect gambling content, such as the use of BERT or a combination of TF-IDF with Random Forest, which achieves accuracy up to 98% in Indonesian language text. The development of Large Language Models (LLMs) such as Llama 3.2 opens new opportunities due to their ability to process more complex natural language and their efficiency through PEFT and LoRA fine-tuning techniques. The support of the AdamW optimizer has also been proven to improve training stability and model generalization.

This study focuses on the implementation of Llama 3.2, which is fine-tuned using Hugging Face to detect Indonesian-language online gambling promotional texts, as well as evaluating the model's performance without an optimizer and with an optimizer. The results are expected to have implications for the development of automatic detection systems for negative content, particularly the promotion of online gambling.

II. LITERATURE REVIEW

A. Related Research

Previous studies have demonstrated the significant potential of Large Language Models (LLMs) for detecting harmful content. Nguyen et al. [2] successfully used Llama2-7B to detect sexual predators and abusive language with near perfect accuracy. Lv et al. [3] developed a LoRA-based T-Llama for the Tibetan language with superior results in text classification. Oliveira et al. [4] demonstrated that Llama 3.1 8B can effectively detect toxic text in Portuguese through prompt optimization. Zhao et al. [5] utilized Llama3-70B for multi-task learning, surpassing RoBERTa's performance in general classification. Repede and Brad [6] adapted Llama 3–8B for multilingual hoax detection with better performance than baseline models.

In addition, Smith et al. [7] applied BERT to detect signs of gambling problems on German-language online forums, while Perdana et al. [8] used a combination of TF-IDF and Random Forest to detect online gambling promotions in Indonesian with an accuracy up to 98%.

From these various studies, it can be seen that although

Received: Jan. 28, 2026; received in revised form: Oct 30, 2026; accepted: Oct. 30, 2026; available online: Oct. 30, 2026.

Corresponding: mrafid.mf@upi.edu

In Press

Copyright © 2026

LLMs have proven to be effective in various domains, specific research related to the detection of online gambling promotions in Indonesian using Llama 3.2 has not been widely conducted, thus becoming a research gap that this study aims to address.

B. Online Gambling

Online gambling is a form of internet-based gambling that can be accessed through digital devices such as computers or mobile phones. This activity poses risks of addiction, financial loss, and potential criminality [9], [10]. Based on Financial Transaction Reports and Analysis Center (PPATK) shows that online gambling transactions in Indonesia in 2023 reached more than Rp200 trillion, with more than 2 million active players, including teenagers and adults [11].

Online gambling has numerous types, such as poker, lottery, virtual casinos, slots, soccer betting, and digital competition-based games. Their popularity is increasing due to aggressive promotional strategies, including the use of public figures and influencers [12], advertising on social media [13], links to major sporting events [14], and bonus offers and referral programs [15].

These factors indicate that online gambling is not merely digital entertainment, but rather a socio-economic phenomenon that requires serious attention and technology-based solutions.

C. Large Language Models (LLM)

Large Language Models (LLMs) are deep learning algorithms trained with large-scale data to understand, summarize, translate, and generate natural text. These models use a transformer architecture that enables the understanding of complex language patterns and have demonstrated superior performance in various Natural Language Processing (NLP) tasks such as translation, text generation, and question answering [16], [17].

The development of LLM since 2017 has been marked by the emergence of GPT, Llama, and their derivatives, which are capable of generating human-like language. LLM is now widely applied in various domains, including education, mental health, software engineering, and recommendation systems [18], [19]. Their advantages include language comprehension and generation capabilities, broad cross-domain applicability, and reasoning and generalization abilities [20], [21].

However, LLMs also have limitations, including high computational costs, potential bias in data, limitations in handling factual knowledge, and ethical and privacy issues [22], [23], [24]. Therefore, despite their great potential, the use of LLMs requires a prudent approach to optimize their benefits without causing negative impacts.

D. Model Llama3

The Meta AI Large Language Model (Llama 3) is a series of open-source large language models developed by Meta, with native multilingual support, coding, reasoning, and tool usage. Llama 3 is available in various versions (Llama 3.1, 3.2, 3.3) with parameter sizes ranging from 1B to 405B and has demonstrated competitive performance on par with leading models such as GPT-4 on various tasks [25].

The main advantage of Llama 3 is its open nature, which facilitates its use for professional and academic needs, including retraining and fine-tuning. This flexibility makes Llama 3 one of the most promising LLMs for the development of harmful content detection applications.

E. Hugging Face

Hugging Face is one of the major players in the development of artificial intelligence (AI) and Natural Language Processing (NLP) through the provision of open-source libraries and tools. Its flagship product, the Transformers library, has revolutionized the application of modern language models in various fields such as healthcare, finance, customer service, and education [26].

Additionally, Hugging Face Hub (HFH) provides a platform for sharing and managing thousands of cross-domain pre-trained models, including language, computer vision, and audio. HFH supports global community collaboration and facilitates the use and modification of AI models. However, challenges related to documentation transparency, dataset bias, and licensing remain a concern [27], [28].

F. Optimizer

In machine learning, optimizer is an algorithm that iteratively updates model weights to minimize the loss function. An optimizer has many functions that include establishing the learning rate; providing stability; increasing the speed of convergence; and improving the performance of the model on novel data. An optimizer is extremely important to establish the rate at which the algorithm will converge, maintain the stability of the training process, and enhance the ability of the model to generalize [29].

Many different optimization algorithms (optimizers) have been utilized for training Large Language Models (LLMs). These include but are not limited to, AdamW, LAMB, and various versions of Stochastic Gradient Descent (SGD). All of these optimization algorithms have unique advantages in regard to their memory usage and ability to generalize [30]. Out of all the aforementioned optimization algorithms, AdamW is typically viewed as being the best choice when performing "fine tuning" operations on large models like Llama 3. The reason being, AdamW is designed specifically to allow for separate updates to the weights of the model versus applying weight decay. Separating the two allows for improved training stability during the operation and ultimately results in the model having improved generalization capabilities [29].

G. Fine Tuning

Fine-tuning refers to modifying an already trained model with a much smaller, yet more relevant new dataset, which will allow the model to better perform on a particular task. There are three types of fine-tuning: full fine-tuning, partial fine-tuning and PEFT (Parameter Efficient Fine-Tuning) [31], [32].

PEFT enables the fine-tuning of Large Language Models (LLMs) with fewer parameters, thus being more cost-effective in resource consumption and processing time [33]. Among various techniques, Low-Rank Adaptation (LoRA) has gained

In Press

popularity. It involves adding a limited number of additional parameters to each layer of the model. Using LoRA, Llama 3.2 can be efficiently trained on relatively low-resource hardware, and yet maintain high accuracy for detecting specific items [34],[5].

III. Proposed Methods

The purpose of this study was to show how to fine-tune Llama 3.2 1B using Low Rank Adaptation (LoRA), which allows users to train their models in a way that is less expensive than traditional methods of training. This type of training can be done on hardware with small amounts of RAM. The fine-tuning process was done using Hugging Face's Transformers library. The first step in this process was tokenizing the input text using Hugging Face's AutoTokenizer. After the input text was tokenized, it was split into three groups: train, validation, and testing. A method for masking keywords was developed so that the model would not simply rely on surface-level keywords to make its predictions. Instead, the model should be able to identify patterns from the context of the sentence. An example of this is if a user wants to find examples of someone saying they want to gamble, the user could mask the word "gamble" so that the model must look at the context of the sentence to determine if the user is referring to gambling. The model was optimized using the AdamW optimizer during training. The AdamW optimizer was chosen because it provides greater stability when optimizing model weights and also performs better when optimizing large-scale models. When LoRA and the AdamW optimizer are combined, fine-tuning can occur with very little resource consumption and still produce high-quality results. Following the training phase, the model was tested using a Confusion Matrix along with other metrics including Accuracy, Precision, Recall, and F1 Score. Lastly, the model was saved in.zip format for use in an automated detection system.

IV. Experimental Result

A. Collecting Data

The data used in the current research was collected through the Kaggle platform (an open-source data-sharing platform), where it had been uploaded by a Kaggle user named yaemico. It included comments regarding online gambling promotions. In the present study, I have focused on only two primary columns: "message" and "label." These two columns were selected as being the most applicable to the goal of identifying and labeling comments related to online gambling promotions. My selection of these columns follows the methodology employed in Perdana et al.'s [8] study, which identified online gambling promotions in Twitter Indonesia based solely on text content and classification labels. In addition, Rahul et al. [36] used the same approach when they created a text classification model utilizing only two main columns: "comment text" and "label," to differentiate between positive (1) and negative (0) labels.

The first part of the modeling process involves transforming the raw data into a form that can be used within the model. Converting the data is done as a "pre-processing" step. During the pre-processing phase, the data is normalized and cleaned so it has similar properties to other datasets being processed. Also, during this phase four new variables are created. These include: (1) each comment's original message content; (2) each comment's text after "case_folding" (all characters have been converted to lower case); (3) each comment's text after "cleaning_rafid" (all emoticons and special characters have been deleted); and (4) each comment's tag category variable (this variable can contain either of two possible values; 0 indicates comments that do not pertain to online gambling promotions and 1 indicates comments that do pertain to online gambling promotions). At the end of the pre-processing phase there were 6,348 records remaining in our dataset.

The final group of attributes mentioned above will be removed from this study because either they do not relate directly to developing the classification algorithm, or they have already been processed (i.e., clean_rafid and label) and may be removed so that we may produce a reduced size of the original data set. If an attribute was not processed prior to removal, it too will be removed. The final data set used in the analysis will consist of only two main columns: "clean_rafid," which represents the processed comments; and the classification labels associated with those comments.

B. Data Visualization

Visualization and interpretation of results were performed to provide an overview of the dataset content, which was displayed using pie charts, bar charts, and histograms. In the pie chart shown in Fig.1, the distribution of positive and negative labeled posts is clearly visualized. Label 1 represents posts containing online gambling promotions, while label 0 indicates non-gambling posts. The visualization results show that class 0 has a proportion of 49.8% and class 1 has a proportion of 50.2%. This very small difference confirms that the dataset is balanced, so that the trained model has a fair chance of predicting both classes. This balance also minimizes the risk of bias and allows evaluation of metrics such as accuracy, precision, recall, and F1-score to be used more representatively without the need to apply additional balancing methods.

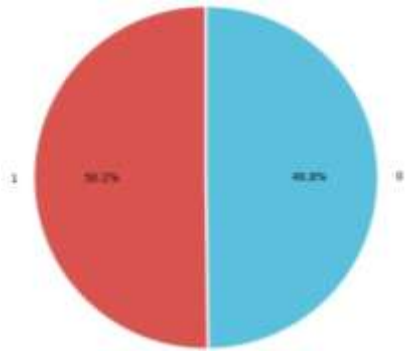


Fig.1. Label Distribution Pie Chart

The next visualization is a bar chart showing how often certain words and bigrams occur most frequently. Figures 2 and 3 illustrate the frequencies of the words and bigrams used in posts labeled 1 (gambling). Results indicate that the most commonly occurring words include *100rb, di, wisdomtotofire*, *ketik*, and *freebet*; whereas the most frequently occurring bigrams include *ketik di, freebet 100rb, di googlewisdomtoto*, and *100rb gratis*. Also, some other new terms such as pas4d, bwtogel, and gacorr were identified as being used by online

In Press

gambling promoters, indicating different promotional methods.

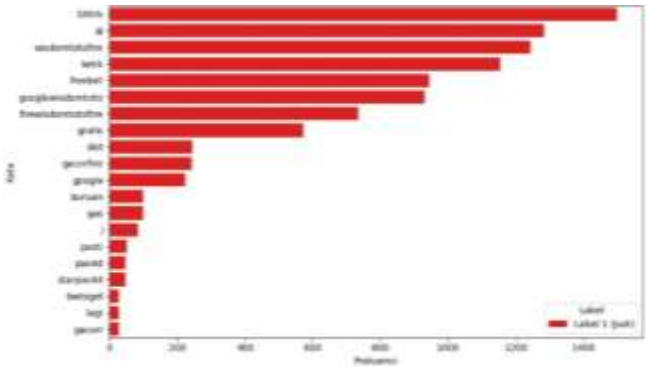


Fig.2. *Bar Chart of Word Frequency – Label 1*

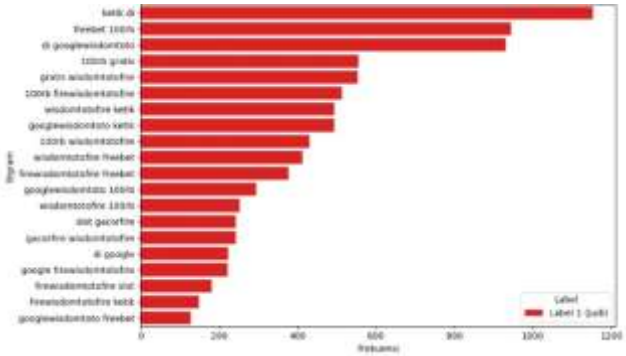


Fig.3. *Bar Chart of Bigram Frequency – Label 1*

In contrast, Fig.4 and Fig.5 show bar charts of word and bigram frequencies in posts labeled 0 (non-gambling). Frequently appearing words include di, jogja, fireangka, terbalik, and bayarfire, while the dominant bigrams are fireangka terbalik, terbalik di, and di bayarfire. In addition, phrases with a celebratory tone also appear, such as ulang tahun, selamat ulang, and hbd jogja. This indicates that non-gambling posts tend to be informative, personal, or related to socio-cultural contexts, which is very different from the pattern of gambling promotion.

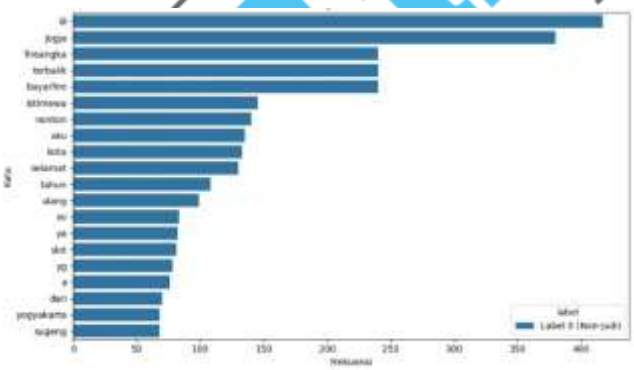


Fig.4. *Bar Chart of Word Frequency – Label 0*

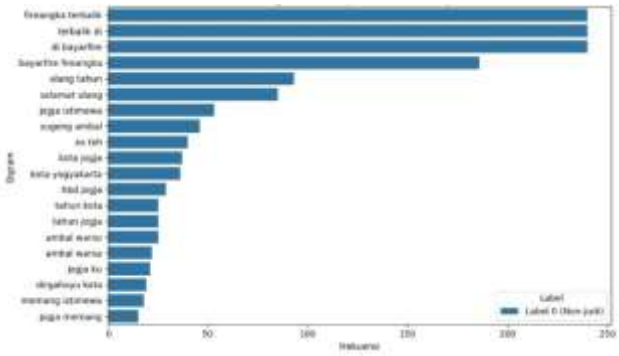


Fig.5. *Bar Chart of Bigram Frequency, Labelled 0*

The final visualization in the form of a histogram in Fig.6 shows the distribution of text length in number of words based on label categories. The results show that posts with label 1 (gambling) are generally shorter, with the majority consisting of only three words, while posts with label 0 (non-gambling) are more varied, with the number of words reaching more than thirty. This pattern confirms that gambling promotions on social media tend to be concise and compact to quickly attract attention, while non-gambling posts have a diversity of structure and message content. This difference in text length distribution can be used as a predictive feature in the automatic detection process, as it provides a clear signal regarding the tendency of the post content.

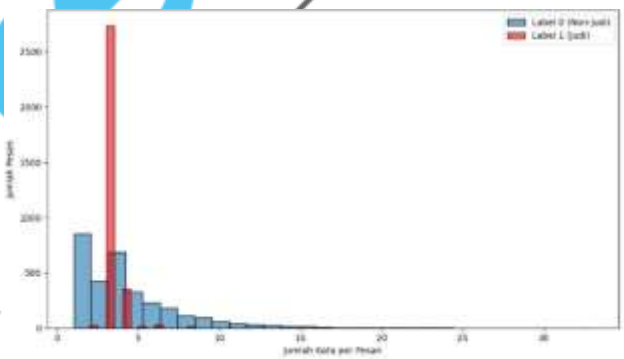


Fig.6. Histogram by Label

C. Pre-Processing Data

This study utilized three primary phases of data pre-processing; case folding, cleaning, and filtering. The processes were completed with the use of the Kaggle dataset, through the Google Colab platform, with the assistance of several different Python libraries. The overall goal of the pre-processing phase was to create a consistent format for the text, clean up the text, increase the quality of the data, increase the accuracy of the models, and reduce potential biases or interferences created by irrelevant tokens.

The initial step of case folding involves changing every letter within a string into its lowercase equivalent. The purpose of case folding is to make sure that the system doesn't treat two different versions of the same word (e.g., "FREEBET" and "freebet") differently because of capitalization. As shown in Table I, the result of case folding can be seen as having transformed the entire string into lowercase letters, thus creating consistency for further analysis.

TABLE I. Case Folding Resuts

Before Case Folding	After Case Folding
Tribun Tv Mana Ini Kenapa Acaranya Belum Dimulai Ini Wayang Jogja Night Carnival	*tribun tv mana ini kenapa acaranya belum dimulai ini wayang jogja night carnival*
SLOT GACOR WISDOMTOTO	*slot gacor wisdomtoto*
FREEBET 100RB WISDOMTOTO	*freebet 100rb wisdomtoto*

The second phase of the process is "cleaning" where you remove unneeded items from the analysis. According to Mohammadi [37], if there are too many non-informative tokens in the analysis, like; emojis, excessive punctuation, usernames, double spacing, numbers used alone, or text-based emojis, then this could negatively affect how well the model works and how much it generalized. Therefore, during the "cleaning" phase you will want to get rid of all of these types of items from the text. Table II illustrates some of the differences in the data before and after cleaning. An example of a sentence that had an emoji: loudly_crying_face: was cleaned to eliminate that item. Also, any punctuation, usernames, isolated numbers, or double spaces were eliminated from the sentences to make them less complex and more focused on the primary topic.

TABLE II. Cleaning Results

Cleaning Steps	Before Cleaning	*After* Cleaning
Deleting emoji	*aku boleh terbang ga sih? biar bisa liat: loudly_crying_face: loudly_crying_face: loudly_crying_face:*	*aku boleh terbang ga sih biar bisa liat*
Deleting punctuation	*"Padahal tugu deket bgt sm rumah tp ga kebagian tempat, kayak tahun lalu."*	*padahal tugu deket bgt sm rumah tp ga kebagian tempat kayak tahun lalu*
Deleting tag username	*@rita setiyaningsih sabar*	*rita setiyaningsih sabar*
Deleting double space	*ga bisa nonton disana, dilive aja:grinning_face_with_big_eyes:*	*ga bisa nonton disana dilive ajaa*
Deleting stand-alone numbers	*acara intinya nanti jam 19.00 ya gaes ya*	*acara intinya nanti jam ya gaes ya*
Deleting emoji text	*macet enggak mas? grinning_face_with_sweat:*	*macet enggak mas*

The third step is called filtering. Filtering is the act of eliminating words that are of no value to the analysis, or what are referred to as stopwords. The stopword list being utilized comes from the stopwords.iso repository and includes some other words that were found to be less useful than others through word cloud frequency analyses. For example, "wah mantap banget tuh" after filtering would be "mantap", and "penonton sudah banyak yg tumbang min" would become "penonton tumbang", as indicated in Table III.

TABLE III. Filtering Results

Before Filtering	After Filtering
udah lewat menit ni	*Menit ni*
wah mantap banget tuh	*mantap*
penonton sudah banyak yg tumbang min	*penonton tumbang*

Additionally, the filtering results are visualized in the form of a word cloud as shown in Fig.7. This visualization illustrates the words that appear most frequently in the dataset, where the size of the words is directly proportional to their frequency. Based on the word cloud, words such as freebet, 100rb, gratis, wisdomtotofire, googlewisdomtoto, and ketik appear as dominant words. These words reveal promotional or advertising patterns, particularly related to online gambling, for example through bonus offers (freebet, 100rb, gratis) and instructions to search for specific sites using the words google or ketik. Thus, the pre-processing stage not only produces cleaner data, but also provides initial indications of the language patterns frequently used in online gambling promotions.

Fig.7. Word Cloud Based on Word Frequency

D. Data Validation

Dataset validation is carried out to ensure that the data used in the research is of sufficient quality to be applied in the modeling stage. The validation process includes several steps: determining whether the data aligns with the research objectives, ensuring that it does not contain irrelevant components, identifying potential redundancies, and evaluating if any relevant pieces of information may be absent. After these steps are completed, researchers can assess whether the dataset meets the requirements for use in modeling.

A review of the structure of the dataset showed that it contained 6,348 records and each record contained two major variables: cleaning_rafid and label. Cleaning_rafid contains a processed version of the initial text in string format, while label is a numeric variable used to indicate the binary class of each record (0/1). Describing the samples simply using basic descriptive statistics showed that the classes were distributed almost equally throughout all samples, with a mean label value of 0.502; therefore, there would be less of a possibility of bias occurring during the training of the models.

After the first cleaning phase, all of the texts were evaluated to identify and eliminate undesirable components such as emoticons/emojis, symbols, usernames, isolated numbers, emoji representations, and excess white space. Since

In Press

none of those components appeared in the evaluation of the texts, it can be concluded that the initial cleaning phase was effective in removing those components. Therefore, the data set has been prepared for the subsequent stage of processing.

At the time of checking for redundancies in the 6,348 data entries, 3,818 entries were found to be duplicates. Although the presence of duplicate data in the context of online gambling promotional activities does not necessarily imply a fault in the system, it may reflect some of the operational procedures used by online gambling operators. Such operational procedures include, but are not limited to, sending out spam messages, utilizing automated systems to send messages, and conducting large-scale mailing campaigns. Research indicates that keeping duplicate data can increase the external validity of the sample group and therefore provide a basis for generalizing recurring patterns. As such, the decision-making process concerning whether to eliminate or retain duplicate data should be based upon the research goals.

In the final stage of the validation process, we will determine if we have any missing values. Based on our analysis results, we did not find any missing values in the two main columns (cleaning_rafid) and (label). All rows had data, so there were no empty rows in either column. As such, the dataset is completely free from issues related to missing values. This means the validation process has shown that the dataset you wish to use is acceptable enough for you to start the modeling phase.

E. Developing Detection Model

The process of applying the Llama 3.2 1B-based Large Language Model (LLM) to detect online gambling promotional text was carried out through several systematic stages. The first stage was searching for relevant datasets, namely comment data containing online gambling promotional content. The datasets obtained then underwent a series of pre-processing steps to improve data quality, including case folding, cleaning, and filtering. After the pre-processing process is complete, data validation is carried out to ensure that the dataset is clean, consistent, and ready for use in the modelling stage.

During the modeling stage, the dataset is divided into three parts, namely the training set, validation set, and test set, with a distribution ratio of 70%, 20%, and 10%, respectively. The 70% training data is used to build the model and learn the patterns in the data, so that the model has good generalization capabilities [40], [41]. The 20% validation data is used to adjust hyperparameters and minimize the risk of overfitting during the training process. This proportion is considered sufficient to evaluate the model's performance periodically without significantly reducing the amount of training data [42]. Meanwhile, 10% of the test data is used to evaluate the final performance of the model on completely new data, where this proportion is considered sufficient to measure the model's performance objectively [40], [41].

To improve modelling efficiency, optimization methods were used in the form of Parameter Efficient Fine-Tuning (PEFT), which aims to accelerate the training process, and Low-Rank Adaptation (LoRA), which can reduce device memory usage. In the final stage, the model is evaluated using metrics based on a confusion matrix and classification report that includes accuracy, precision, recall, and F1-score. These evaluation results are then used to assess the extent to which the model is able to effectively detect online gambling promotional text based on the context that has been learned.

F. Modelling

At the modeling step, we masked our dataset in order to avoid the overly strong influence of explicit terms that could lead to an excessive number of impulses being generated. According to studies conducted by Russell et al. [43], text messages and emails sent out by online casinos/gambling operators frequently include offers of bonuses, cashbacks, and free bets. These incentives greatly enhance the probability of a user engaging in gambling activities shortly after viewing the communication. Additionally, recent research conducted by Xu et al. [44] regarding the development of text-based illegal gambling website identification models has indicated that utilizing masking methods for explicit terms (e.g., “freebet,” “deposit,” “cuan”) enhances the ability of the model to comprehend the overall context of the text rather than solely focusing on individual words. Thus, the application of masking enhances the generalizability and performance of the model, particularly when applied to datasets having differing characteristics. Consequently, this investigation also employs masking for specific words contained within the dataset; including but not limited to “100rb,” “wisdomtotofire,” “googlewisdomtoto,” “freebet,” “slot,” “gacorfire,” “gratis,” “bayarfire,” “fireangka,” “ketik,” “bonus,” “deposit,” “cuan,” and “gacor.”

After the masking process, the dataset that has gone through the pre-processing stage is divided into three subsets, namely the training set, validation set, and testing set. Next, the dataset is converted into Hugging Face Dataset format using the Dataset.from_pandas() function, after first resetting the index to avoid index inconsistencies during training. Tokenization is performed using AutoTokenizer from the Meta Llama 3.2 1B model with token padding settings matching the end-of-sequence token. The model used is AutoModelForSequenceClassification with the parameter num_labels=2, because this research problem is binary classification. Before the fine-tuning process, the model is prepared with the prepare_model_for_kbit_training() function to optimize memory and computation usage with low precision.

The fine-tuning process was carried out by adopting the Parameter-Efficient Fine-Tuning (PEFT) concept using the Low-Rank Adaptation (LoRA) method. With this approach, only about 10% of the model parameters were trained, making training more efficient without sacrificing prediction performance [45]. The LoRA configuration in this study sets the rank to 16, lora_alpha to 32, and the target module on query projection and value projection. Dropout value of 0.05 is applied as a form of regularization to prevent overfitting. Bias is not included in the adaptation process, and the task type is set as sequence classification. This configuration is chosen to balance resource efficiency requirements with the resulting prediction performance.

Next, the trainer configuration is set using the Trainer API from Hugging Face. The main parameters used include a small batch size (2) for training and evaluation, gradient

In Press

accumulation steps of 8 to overcome memory limitations, and a total of 3 epochs with a learning rate of 1e-6 using a cosine scheduler. The warmup ratio is set to 0.1, while the evaluation and model saving processes are performed at each epoch. To ensure efficiency, the best model is automatically selected based on the accuracy metric, and the training results are saved with a maximum of only one model. In addition, calculations are performed using bfloat16 (bf16) precision for greater stability, with load_best_model_at_end enabled to ensure the best performance at the end of training.

This study also tested the use of the AdamW optimizer, which is designed to improve training stability and accelerate convergence by applying decoupled weight decay. Recent research by Selvakumari and Durairaj [46] shows that AdamW provides higher accuracy and lower validation loss compared to other optimizers, especially in large-scale models such as Llama 3.2. In this configuration, AdamW was used with a learning rate of 1e-6 and weight decay of 0.01.

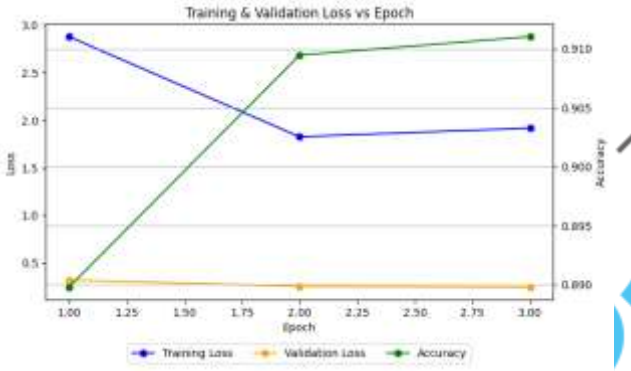


Fig.8. Results of Modelling Without Optimizer

The training results show Fig.8 that in the model without an optimizer, the training loss dropped significantly from around 2.9 to 1.9 in three epochs, while the validation loss remained stable at around 0.3. The validation accuracy increased from 89% to 91% consistently, indicating that the model was able to learn data patterns well without overfitting or underfitting.

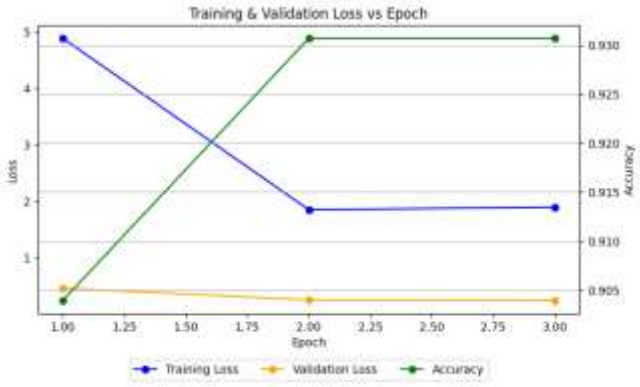


Fig.9. Results of Modelling with Optimizer

On the other hand, the model with the AdamW optimizer was shown to be much better at learning than the previous model as can be seen in Fig. 9. Both models were able to learn from the data; however, the model with the AdamW optimizer was able to learn faster and more effectively than the other. The model with the AdamW optimizer had a training loss of 4.90 initially and then dropped to 1.88 by the end of the first epoch. It also had a validation loss of 0.27 by the end of the first epoch. Additionally, the model with the AdamW optimizer had a validation accuracy that increased significantly faster than the other model. At the end of the second epoch the validation accuracy for the model with the AdamW optimizer reached 93% and then stabilized during the third epoch. These findings demonstrate that using AdamW will allow for greater learning efficiency while still providing similar generalization capabilities on validation data. Therefore, although both models were able to successfully identify the pattern within the data, the model with an optimizer provided more efficient and stable results.

G. Evaluating Model Performance

The evaluation of model performance on the test data is performed using two primary metrics: the Classification Report and the Confusion Matrix. The Classification Report metric measures the precision, recall, F1-score, and accuracy values for all classes [47]. On the other hand, the Confusion Matrix allows us to compare the predicted class with the true class, which enables us to understand the error types made by the classifier [48]. Therefore, these metrics are complementary when evaluating model performance.

TABLE IV. Both Model Classification Report

Model	Label	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Without optimizer	Non-Judi	0.90	0.87	0.88	0.89
	Judi	0.87	0.91	0.89	
With optimizer	Non-Judi	0.89	0.94	0.92	0.92
	Judi	**0.94**	0.89	**0.92**	

The testing was conducted using 635 test entries, comparing two approaches: a fine-tuned model without an optimizer and a fine-tuned model with an optimizer on Llama 3.2 variant 1B. The classification report evaluation results show that in the model without an optimizer, the non-gambling class has a precision value of 0.90, a recall of 0.87, and an F1-score of 0.88. Meanwhile, the gambling class obtained a precision value of 0.87, a recall of 0.91, and an F1-score of 0.89, with an overall accuracy value of 0.89. These results show that the model is quite capable of classifying both classes well, although there are still several errors in distinguishing between non-gambling and gambling data.

After applying the optimizer, the model's performance improved significantly. For the non-gambling class, precision decreased slightly to 0.89, but recall increased sharply to 0.94, resulting in an F1-score of 0.92. As for the gambling class, precision increased to 0.94 with a recall of 0.89, resulting in an F1-score of 0.92. Overall, the accuracy value increased to 0.92. This improvement shows that the application of the optimizer made the model more balanced in detecting both classes, with a better balance between precision and recall.

Analysis through the confusion matrix reinforces these findings. In the model without an optimizer, out of a total of 635 test data, 274 non-gambling data were correctly classified, but there were still 42 non-gambling data that were

misclassified as gambling. For the gambling class, the model was able to classify 290 data correctly, but there were 29 data that were misclassified as non-gambling. This explains why, even though the accuracy value is quite high, the model still shows an imbalance in performance between the two classes, shown Fig.10.

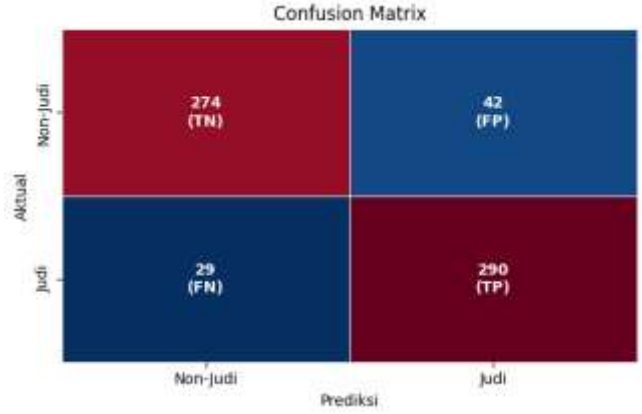


Fig.10. Confusion Matrix Result Without Optimizer

Conversely, in the model with an optimizer, the classification results were more even. Of the total test data, 298 non-gambling data were correctly recognized, while only 18 data were misclassified as gambling. For the gambling class, 284 data were correctly predicted, while 35 data were misclassified as non-gambling. These results indicate an improvement in terms of prediction balance between classes. Although the recall value in the gambling class decreased slightly, the increase in precision and stability of the F1-score proves that the optimizer successfully helped the model learn more effectively, shown Fig.11.

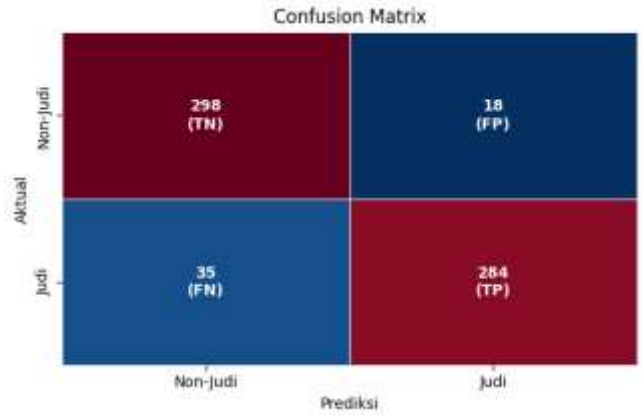


Fig.11 Confusion Matrix Result with Optimizer

Overall, the evaluation results confirm that the application of the AdamW optimizer can significantly improve the performance of the Llama 3.2 1B variant model, both in terms of overall accuracy and prediction balance between classes. Models with optimizers are not only more accurate in classification, but also able to reduce bias towards one class, making them more reliable for use in detecting online gambling promotional text.

H. Model Implementation

After going through the modelling stages, the results of the online gambling promotion text predictions can be seen in full in Appendices 2 and 3 for the model without an optimizer, and in Appendices 4 and 5 for the model with an optimizer. One example of a prediction shows a significant difference. In the sentence “mesti asline macet pol”, the fine-tuned model without an optimizer predicts the text as ‘Gambling’, even though the original label is “non-gambling”, making the prediction incorrect. Conversely, the model with an optimizer can provide the correct prediction, namely “non-gambling”, in accordance with the original label. This difference confirms that the application of an optimizer can help the model produce more accurate classifications.

The total number of predictions generated by both models is equal to the number of test data, which is 635 data. However, to test the generalization ability, testing was also carried out on data outside the dataset. This aims to see the extent to which the model is able to maintain performance on new data that has never been used in the training process. The test results show that models without optimizers still often fail to recognize text that clearly contains elements of online gambling promotion. For example, in the sentences “Latest hot slots 2025, 150% new member bonus and free spins every day” and “REGISTER FOR TODAY'S HOT SLOTS 100% NEW MEMBER BONUS”, the model without an optimizer gave the prediction “non-gambling”, when the original label was “Gambling”. This failure demonstrates that the model is incapable of identifying when there is an explicit promotion of gambling through the use of a pattern-based recognition algorithm. On the other hand, the model that has been properly trained/optimized was successful in identifying every sentence as pertaining to "Gambling".

Both models were able to identify sentences that did not contain any references to gambling as "not gambling." For example, the sentence "Big discounts on online shopping on November 11th, free shipping across Indonesia," and the sentence "This sentence is like using AI; it matches with what he said." The major difference between the two models was that the model that used an optimization algorithm was more accurate at identifying promotional material related to gambling.

The results from evaluating both the test dataset and the non-test dataset show that using optimization techniques can significantly increase the effectiveness of models. Models that were developed with the use of optimization techniques had a much higher success rate in detecting gambling-related content, as well as being able to apply knowledge gained from one dataset to another. Therefore, by applying optimization techniques to models, they will be more capable of adapting to differences in language used in on-line gambling advertising materials; therefore, increasing the reliability of models used in automated detection systems.

I. Comparation of Two Model

As shown by the test results, using an optimizer in the model resulted in a higher level of performance than the model that did not use an optimizer. Performance improvements were observed on almost all metrics, for both non-gambling and gambling labels. On the non-gambling side, the precision dropped slightly (from.90 to.89), while the recall rose significantly (from.87 to.94). Consequently, the F1-score rose from.88 to.92. Thus, the use of the optimizer improved the

Draft

consistency with which the model recognized non-gambling data and enhanced its ability to reduce error in identifying that category. Meanwhile, for gambling labels, precision increased from 0.87 to 0.94, although recall decreased from 0.91 to 0.89. Despite a slight decrease in recall, the F1 score still increased from 0.89 to 0.92. Thus, the balance between precision and recall was maintained, enabling the model to provide more stable predictions for both classes. Overall, the model's accuracy increased from 0.89 to 0.92, indicating that the use of the optimizer not only improved precision but also improved the balance of evaluation between classes, shown Fig.12.

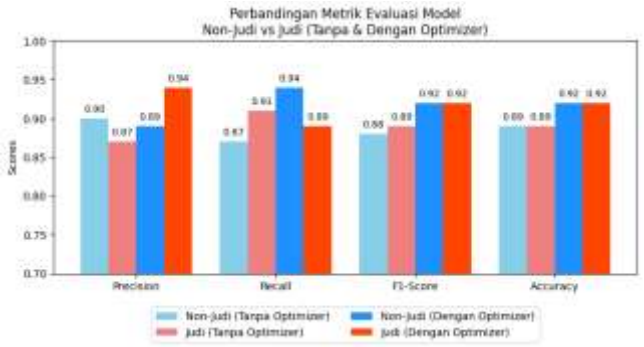


Fig.12 Comparison Diagram of Classification Report Without Optimizer and With Optimizer

The results of the evaluation using a confusion matrix further reinforce these findings. Models with optimizers produced more correct predictions than models without optimizers. This can be seen from the increase in the number of True Negatives from 274 to 298, which means that the model is better at accurately identifying non-gambling text. In addition, the number of errors in the False Positive category decreased significantly from 42 to 18, indicating a reduced tendency for the model to misclassify non-gambling text as gambling. However, there was a slight compromise in gambling class detection, where the number of True Positives decreased from 290 to 284 and False Negatives increased from 29 to 35. Despite a slight decline in gambling content detection performance, consistent improvements in non-gambling detection and a reduction in false positives made the model with the optimizer more balanced and reliable in the overall classification process, shown Fig.12.

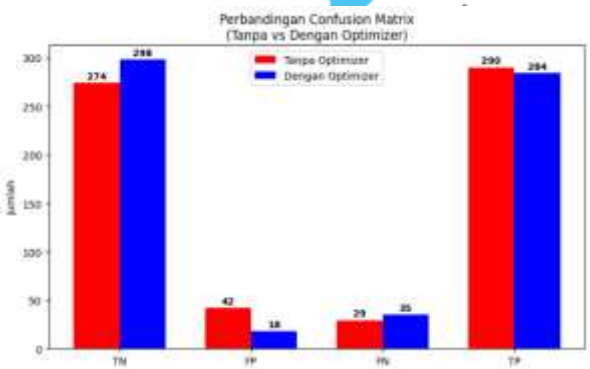


Fig.12. Comparison Diagram of Confusion Matrix Without Optimizer and With Optimizer

In general, this comparison shows that the use of optimizers in the fine-tuning process is effective in improving model stability and quality. Optimizers not only help accelerate convergence but also strengthen the model's generalization capabilities for new data. Thus, optimized models are more suitable for use in real-world applications for detecting online gambling promotional text, as they can reduce classification errors and maintain a balance of performance between the two labels.

V. Conclusion

Based on the results of the research conducted, it can be concluded that the application of the Llama 3.2 variant 1B model successfully classified online gambling promotional content well. The dataset used came from Kaggle with a balanced distribution, namely 3187 posts labeled 1 (50.2%) as online gambling promotions and 3161 posts labeled 0 (49.8%) as non-promotions. The implementation of a fine-tuned model with the AdamW optimizer showed superior performance compared to models without optimizers. The test results showed an accuracy of 92%, with precision of 89%, recall of 94%, and F1-score of 92% in class 0 (non-gambling), as well as precision of 94%, recall of 89%, and F1-score of 92% in class 1 (Gambling). During the training process, the train loss and validation loss values decreased consistently without any indication of overfitting or underfitting, indicating that the model was able to generalize well to the test data.

This study has important implications for the development of automatic detection systems for negative content, particularly online gambling promotion. The fine-tuning process using the Hugging Face library on the Llama 3.2 model proved effective in producing accurate predictions. Evaluation using a confusion matrix and classification report demonstrated that the model was able to accurately identify text patterns in both categories, namely, promotion of gambling and non-promotion. Therefore, this study is expected to contribute to the body of knowledge and practical application of LLM in classifying high-risk text and may serve as a resource for future research on identifying illegal content on social media and other digital channels.

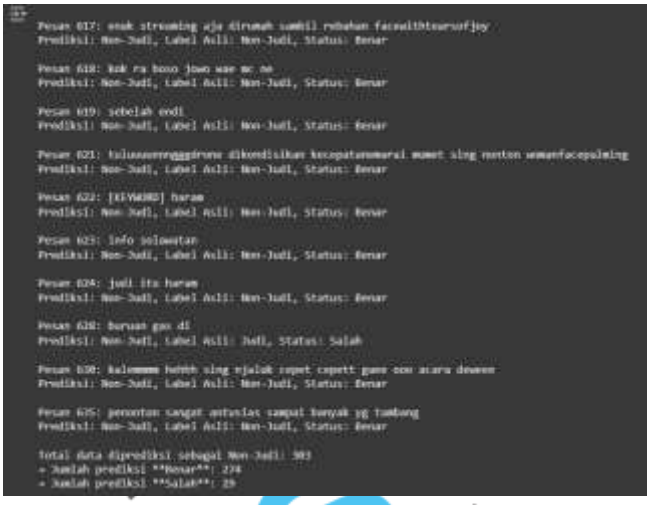
The results of the study indicate that although the model developed in this study performs well, there are several areas where additional work can be done. It is suggested that subsequent studies utilize additional types of LLM models including larger versions of Llama 3.2, and other models such as BERT, Gemma, Phi, or DeepSeek, in order to evaluate their performance relative to one another. Also, future studies will need to use a larger and more diverse dataset containing little to no duplicates in order to allow the model to gain insight into a wider range of online gambling promotion. Other options include testing alternative library platforms (other than Hugging Face) in order to determine if they enable the addition of more parameters to enhance the model's performance; especially when using high-performance computing resources. Lastly, integrating the model's output into actual applications (e.g., browser extensions or websites) designed to identify online gambling promotions would provide opportunities for users/respondents to assess the efficacy of the model in a real-world setting.

In Press

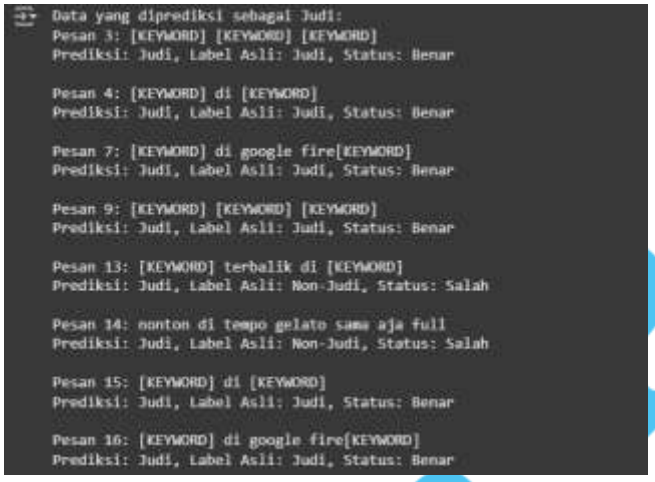
Attachment 1 *Pre-Processing Dataset Results*

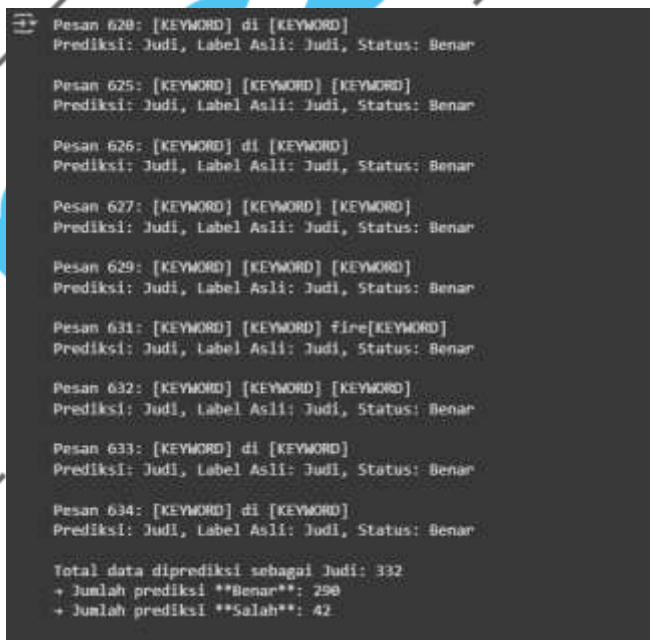
cleaning_rafid,label
assalamualaikum,0
waalaikumussalam,0
udah lewat menit ni,0
wib,0
tribun tv mana ini kenapa acaranya belum dimulai ini wayang jogja night carnival,0
saya nonton apa ini tribun tv kenapa acara wayang tribun night carnival tak kunjung dimulai,0
kapan mulainya min pengen keburu nonton,0
assalamualaikum,0
lama,0
ini nanti bakal live disini taa,0
mash nnt jam 7an hbs isya,0
la yang reog ponorogo tuh kok tulisannya jam ya,0
apa sebagian udah mulai,0
ngko aku tak melu ngebakki dalanbeamingfacewithsmilingeyes,0
sampai kapan min,0
nak irvan pindah ajh dr sebelah,0
selamat ulang tahun ke268 kota yogyakarta rahayu rahayu rahayu,0
nonton dari rumah jalah malas macet2,0
reng ponorogo udah tampil belum ya teman,0
acara intinya nanti jam ya gaes ya,0
selamat ulang tahun jogjaku tercinta terimakasih sudah menjadi kota penuh kenangan,0
jogja memang istimewa,0
otw ke tugu,0
untung rumah ku deket loudlycryingface,0
adzan2,0
trimakasih tribun jogja saya bisa lihat gak desak2 dan gak harus ketugu,0
adzan boss adzan,0
kebagian paling depan loudlycryingface,0
ak bakalan tunggu di sini buat liat wayang loudlycryingface,0
beli jajan dulu laper smilingfacewithtear,0
adzan monggo solat rumiyen,0
parkirnya dimana ya gaes,0
harus y ad videotron biar bisa liat di jauhh,0
muleh wae mas nonton live tribunmacet dan desak desakan,0
aq liat dri sini ja,0
infonya yg dilokasi grinningfacewithsweat,0
yg di lokasi pooolll rame ne,0

Attachment 2 *Results of Online Non-Gambling Label Prediction Without Optimizer*

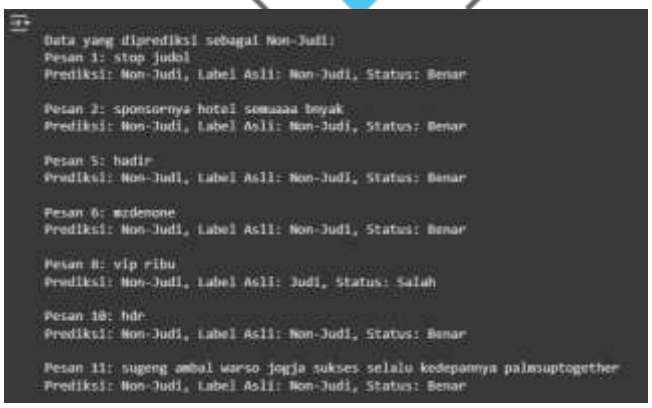


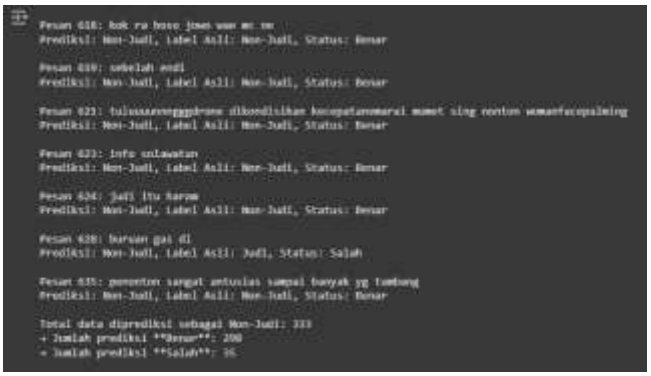
Attachment 3 *Results of Online Gambling Label Prediction Without Optimizer*





Attachment 4 *Results of Online Non-Gambling Label Prediction with Optimizer*





Attachment 5 *Results of Online Gambling Label Prediction with Optimizer*

```
Data yang diprediksi sebagai Judi:
Pesan 3: [KEYWORD] [KEYWORD] [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 4: [KEYWORD] di [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 7: [KEYWORD] di google fire[KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 9: [KEYWORD] [KEYWORD] [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 15: [KEYWORD] di [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 16: [KEYWORD] di google fire[KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 18: [KEYWORD] di [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar
```

```
Pesan 629: [KEYWORD] [KEYWORD] [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 630: kalemmmm hehhh sing njaluk cepet cepett gaee ooo acara deweee
Prediksi: Judi, Label Asli: Non-Judi, Status: Salah

Pesan 631: [KEYWORD] [KEYWORD] fire[KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 632: [KEYWORD] [KEYWORD] [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 633: [KEYWORD] di [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Pesan 634: [KEYWORD] di [KEYWORD]
Prediksi: Judi, Label Asli: Judi, Status: Benar

Total data diprediksi sebagai Judi: 302
- Jumlah prediksi **Benar**: 284
- Jumlah prediksi **Salah**: 18
```

Attachment 6 *Metrics Calculation Research Without Optimizer*

Metrics	Calculation
Precision	$Precision^{Judi} = \frac{TP}{TP + FP} = \frac{290}{290+42} = \frac{290}{332} = 0.87$
	$Precision^{Non\text{-}Judi} = \frac{TN}{TN + FN} = \frac{274}{274+29} = \frac{274}{303} = 0.90$
Recall	$Recall^{Judi} = \frac{TP}{TP + FN} = \frac{290}{290+29} = \frac{290}{319} = 0.91$
	$Recall^{Non\text{-}Judi} = \frac{TN}{TN + FP} = \frac{274}{274+42} = \frac{274}{316} = 0.87$
F1-*Score*	$F1\text{-}score^{Judi} = 2 \text{ x } \frac{Precision \text{ X } Recall}{Precision + Recall} = 2 \text{ x } \frac{0.87 \text{ x } 0.91}{0.87+0.91} = 2 \text{ x } \frac{0.7917}{1.78} = 0.89$
	$F1\text{-}score^{Non\text{-}Judi} = 2 \text{ x } \frac{Precision \text{ X } Recall}{Precision + Recall} = 2 \text{ x } \frac{0.90 \text{ x } 0.87}{0.90+0.87} = 2 \text{ x } \frac{0.783}{1.77} = 0.89$
Accuracy	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{290 + 274}{290 + 274 + 42 + 29} = \frac{564}{635} = 0.89$

Attachment 7 *Metrics Calculation Research with Optimizer*

Metrics	Calculation
Precision	$Precision^{Judi} = \frac{TP}{TP + FP} = \frac{284}{284+18} = \frac{284}{302} = 0.94$
	$Precision^{Non\text{-}Judi} = \frac{TN}{TN + FN} = \frac{298}{298+35} = \frac{298}{333} = 0.89$
Recall	$Recall^{Judi} = \frac{TP}{TP + FN} = \frac{284}{284+35} = \frac{284}{319} = 0.89$
	$Recall^{Non\text{-}Judi} = \frac{TN}{TN + FP} = \frac{298}{298+18} = \frac{298}{316} = 0.94$
F1-*Score*	$F1\text{-}score^{Judi} = 2 \text{ x } \frac{Precision \text{ X } Recall}{Precision + Recall} = 2 \text{ x } \frac{0.94 \text{ x } 0.89}{0.94+0.89} = 2 \text{ x } \frac{0.8366}{1.81} = 0.92$
	$F1\text{-}score^{Non\text{-}Judi} = 2 \text{ x } \frac{Precision \text{ X } Recall}{Precision + Recall} = 2 \text{ x } \frac{0.89 \text{ x } 0.94}{0.89+0.94} = 2 \text{ x } \frac{0.8366}{1.81} = 0.92$
Accuracy	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{284 + 298}{284 + 298 + 18 + 35} = \frac{582}{635} = 0.92$

In Press

REFERENCES

[1] PPATK, *Laporan Tahunan 2023*, Jakarta: Pusat Pelaporan dan Analisis Transaksi Keuangan, 2024. [Online]. Available: https://www.ppatk.go.id

[2] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "Improving Language Model Pretraining with Word Structural Knowledge," *arXiv preprint*, arXiv:2402.05932, 2024.

[3] Y. Lv, Z. Xie, H. Liu, and Y. Yang, "Efficient Fine-Tuning of Large Language Models via Parameter-Efficient Low-Rank Adaptation," *Knowledge-Based Systems*, vol. 285, p. 111298, 2024.

[4] R. Oliveira, J. Silva, and M. Martins, "Large Language Models in Financial Applications: Opportunities and Risks," in *Proc. 2024 Int. Conf. Artificial Intelligence Applications*, pp. 102–112.

[5] J. Zhao *et al.*, "LoRA Land: 310 Fine-tuned LLMs that Rival GPT-4, A Technical Report," *arXiv preprint*, arXiv:2405.00732, 2024, doi: 10.48550/arXiv.2405.00732.

[6] S. E. Repede and R. Brad, "Llama 3 vs. State-of-the-Art Large Language Models: Performance in Detecting Nuanced Fake News," *Computers*, vol. 13, no. 11, 2024.

[7] E. Smith, J. Peters, and N. Reiter, "Automatic detection of problem-gambling signs from online texts using large language models," *PLOS Digital Health*, vol. 3, no. 9, p. e0000605, 2024.

[8] R. Perdana, A. Budi, I. Santoso, A. Ramadiah, and P. Putra, "Detecting Online Gambling Promotions on Indonesian Twitter Using Text Mining Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 8, 2024, doi: 10.14569/ijacsa.2024.0150893.

[9] L. Rafiqah and H. Rasyid, "Dampak Judi Online terhadap Kehidupan Sosial Ekonomi Masyarakat," *Al-Mutharahah: J. Penelitian dan Kajian Sosial Keagamaan*, vol. 20, no. 2, pp. 282–290, 2023.

[10] H. H. Zain, M. F. Ramadhan, F. Saputra, and T. Supriyadi, "Pengaruh judi online terhadap kecenderungan remaja melakukan tindakan kriminal," *Triwikrama: J. Ilmu Sosial*, vol. 6, no. 8, pp. 41–50, 2025, doi: 10.6578/triwikrama.v6i8.9835.

[11] CNBC Indonesia, "PPATK: 78% Penjudi Online Berpenghasilan di Bawah Rp 100 Ribu," *CNBC Indonesia*, Sep. 25, 2023. [Online]. Available: https://www.cnbcindonesia.com/news/20230925115129-8-475252

[12] M. F. Pradika, L. T. T. Hidayat, A. H. D. Prakoso, and A. A. Khan, "Study of Online Gambling Promotion Policy in Indonesia, Pakistan, and USA," *Ettisal: J. Communication*, vol. 9, no. 1, 2024.

[13] M. Guillou-Landreat, K. Gallopel-Morvan, D. Lever, D. Le Goff, and J. Y. Le Reste, "Gambling marketing strategies and the internet: What do we know? A systematic review," *Front. Psychiatry*, vol. 12, p. 583817, 2021.

[14] A. E. Killick and M. D. Griffiths, "A content analysis of gambling operators' Twitter accounts at the start of the English Premier League football season," *J. Gambling Studies*, vol. 36, no. 1, pp. 319–341, 2020.

[15] I. Y. Nono, A. A. S. L. Dewi, and I. P. G. Seputra, "Penegakan Hukum Terhadap Selebgram yang Mempromosikan Situs Judi Online," *J. Analogi Hukum*, vol. 3, no. 2, pp. 235–239, 2021.

[16] M. Raiaan *et al.*, "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024, doi: 10.1109/ACCESS.2024.3365742.

[17] L. Fan, L. Li, Z. …, S. Lee, H. Yu, and L. Hemphill, "A Bibliometric Review of Large Language Models Research from 2017 to 2023," *ACM Trans. Intell. Syst. Technol.*, 2023, doi: 10.48550/arXiv.2304.02020.

[18] X. Hou *et al.*, "Large Language Models for Software Engineering: A Systematic Literature Review," *arXiv preprint*, arXiv:2308.10620, 2023, doi: 10.48550/arXiv.2308.10620.

[19] L. Yan *et al.*, "Practical and ethical challenges of large language models in education: A systematic scoping review," *Br. J. Educ. Technol.*, vol. 55, pp. 90–112, 2023, doi: 10.1111/bjet.13370.

[20] S. Minaee *et al.*, "Large Language Models: A Survey," *arXiv preprint*, arXiv:2402.06196, 2025, doi: 10.48550/arXiv.2402.06196.

[21] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, "Large Language Models on Graphs: A Comprehensive Survey," *arXiv preprint*, arXiv:2312.02783, 2024, doi: 10.48550/arXiv.2312.02783.

[22] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying Large Language Models and Knowledge Graphs: A Roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, pp. 3580–3599, 2023, doi: 10.1109/TKDE.2024.3352100.

[23] S. Wang *et al.*, "Knowledge Editing for Large Language Models: A Survey," *arXiv preprint*, arXiv:2310.16218, 2024, doi: 10.48550/arXiv.2310.16218.

[24] S. Bowman, "Eight Things to Know about Large Language Models," *arXiv preprint*, arXiv:2304.00612, 2023, doi: 10.48550/arXiv.2304.00612.

[25] A. Dubey *et al.*, "The Llama 3 herd of models," *arXiv preprint*, arXiv:2407.21783, 2024, doi: 10.48550/arXiv.2407.21783.

[26] U. R. Pol, P. S. Vadar, and T. T. Moharekar, "Hugging Face: Revolutionizing AI and NLP," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 8, pp. 1121–1124, 2024.

[27] A. Ait, J. Izquierdo, and J. Cabot, "On the suitability of hugging face hub for empirical studies," *Empirical Software Engineering*, vol. 30, no. 57, 2025, doi: 10.1007/s10664-024-10608-8.

[28] F. Pepe, V. Nardone, A. Mastropaolo, G. Bavota, G. Canfora, and M. Di Penta, "How do Hugging Face Models Document Datasets, Bias, and Licenses? An Empirical Study," in *Proc. IEEE/ACM Int. Conf. Program Comprehension (ICPC)*, 2024, pp. 370–381.

[29] A. Das, "Natural GaLore: Accelerating GaLore for memory-efficient LLM Training and Fine-tuning," *arXiv preprint*, arXiv:2410.16029, 2024, doi: 10.48550/arXiv.2410.16029.

[30] H. Zhu *et al.*, "APOLLO: SGD-like Memory, AdamW-level Performance," *arXiv preprint*, arXiv:2412.05270, 2025, doi: 10.48550/arXiv.2412.05270.

[31] K. Church, Z. Chen, and Y. …, "Emerging trends: A gentle introduction to fine-tuning," *Natural Language Engineering*, vol. 27, pp. 763–778, 2021, doi: 10.1017/S1351324921000322.

[32] G. Vrbancic and V. Podgorelec, "Transfer Learning With Adaptive Fine-Tuning," *IEEE Access*, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.

[33] A. Hadji-Kyriacou and O. Arandjelović, "Context-PEFT: Efficient Multi-Modal, Multi-Task Fine-Tuning," *arXiv preprint*, arXiv:2312.08900, 2023, doi: 10.48550/arXiv.2312.08900.

[34] A. Tahir, L. Cheng, and H. Liu, "JORA: JAX tensor-parallel LoRA library for retrieval augmented fine-tuning," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2024, pp. 152–159.

[35] S. Ebneyamini, "Towards Developing a Library for Conducting Management Studies Using Design Research Methodology," *Int. J. Qualitative Methods*, vol. 21, 2022, doi: 10.1177/16094069221112245.

[36] R. Rahul, H. Kajla, J. Hooda, and G. Saini, "Classification of online toxic comments using machine learning algorithms," in *Proc. ICICCS*, 2020, pp. 1119–1123, doi: 10.1109/ICICCS48265.2020.9120939.

[37] B. Mohammadi, "Explaining large language models decisions using shapley values," *arXiv preprint*, arXiv:2404.01332, 2024.

[38] P. Lindner, J. Ramnerö, E. Ivanova, and P. Carlbring, "Studying Gambling Behaviors and Responsible Gambling Tools in a Simulated Online Casino," *Front. Psychiatry*, vol. 11, 2021, doi: 10.3389/fpsyt.2020.571954.

[39] H. Syahputra and A. Wibowo, "Comparison of Support Vector Machine (SVM) and Random Forest Algorithm for Detection of Negative Content on Websites," *J. Ilmiah Tek. Elektro Komp. Informatika*, vol. 9, no. 1, 2023, doi: 10.26555/jiteki.v9i1.25861.

[40] H. Bichri, A. Chergui, and M. Hain, "Investigating the Impact of Train/Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 2, 2024, doi: 10.14569/ijacsa.2024.0150235.

[41] T. Thien and W. Yeo, "A comparative study between PCR, PLSR, and LW-PLS on predictive performance at different data splitting ratios," *Chem. Eng. Commun.*, vol. 209, pp. 1439–1456, 2021, doi: 10.1080/00986445.2021.1957853.

[42] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance," *J. Analysis*

In Press

and Testing, vol. 2, pp. 249–262, 2018, doi: 10.1007/s41664-018-0068-2.

[43] A. Russell, N. Hing, M. Browne, and V. Rawat, "Are direct messages (texts and emails) from wagering operators associated with betting intention and behavior? An ecological momentary assessment study," *J. Behavioral Addictions*, vol. 7, pp. 1079–1090, 2018, doi: 10.1556/2006.7.2018.99.

[44] X. Xu, J. Bowen, and R. Taheri, "Token Masking Improves Transformer-Based Text Classification," *arXiv preprint*, arXiv:2505.11746, 2025.

[45] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," *ICLR*, vol. 1, no. 2, pp. 3–15, 2022.

[46] S. Selvakumari and M. Durairaj, "A Comparative Study of Optimization Techniques in Deep Learning Using the MNIST Dataset," *Indian J. Sci. Technol.*, vol. 18, no. 10, 2025, doi: 10.17485/ijst/v18i10.121.

[47] L. Fu, P. Liang, X. Li, and C. Yang, "A machine learning based ensemble method for automatic multiclass classification of decisions," in *Proc. 25th Int. Conf. Evaluation and Assessment in Software Engineering*, 2021, pp. 40–49.

[48] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers & Operations Research*, vol. 152, p. 106131, 2023, doi: 10.1016/j.cor.2022.106131.

In Press