Criminal Court Judgment Prediction System Built on Modified BERT Models

Shannen Latisha, Sean Favian, and Derwin Suhartono *

Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, 11480, Indonesia
Email: shannen.latisha@binus.ac.id (S.L.); sean.favian@binus.ac.id (S.F.); dsuhartono@binus.edu (D.S.)
*Corresponding author

***Abstract*—The high crime rate in Indonesia that occurs from year to year causes a high number of cases that must be examined, tried, and decided through the courts as stipulated in Law No. 48 of 2009 concerning Judicial Power. Therefore, this study was conducted to build a system for predicting sentences resulting from criminal court decisions in the Republic of Indonesia which is expected to facilitate the implementation of jurisprudence. The prediction system was built by comparing 6 Bidirectional Encoder Representations from Transformers (BERT) models and a A Robustly Optimized BERT Pretraining Approach (RoBERTa) model on 3 different proposed architectures: BERT Base, Hierarchical BERT + Mean Pooling, and Hierarchical BERT + LSTM (Long Short-Term Memory). The compared models include *indobert-base-p1, indobert-base-uncased, legal-indobert-indonlu, legal-indobert-indolem, indobert-large-p1, indonesian-roberta-base*. Those models are also compared with Support Vector Machine (SVM)+Term Frequency–Inverse Document Frequency (TF-IDF) as a baseline. The legal-indobert-indolem model with the Hierarchical BERT + Mean Pooling architecture succeeded in performing multi-class classification tasks into 14 classes with the highest F1-score value of 79.8888%. Thus, the successfully created model can be further used in assisting jurisprudence as it has developed the ability to predict criminal court decisions based on similar previously documented cases.**

***Keywords*—court decision prediction, Indonesian criminal court, Bidirectional Encoder Representations from Transformers (BERT) model, Indonesian BERT, hierarchical BERT, legal BERT**

I. Introduction

The majority of nations in the world, including Indonesia, are continually dealing with the issue of crime. According to crime statistics released by Pusiknas Bareskrim Polri (Criminal Investigation Agency of the Indonesian National Police National Criminal Information Center) [1], as depicted in Fig. 1, the number of crimes that took place between 2021 and 2023 grew year over year. In light of the fact that crimes can be perpetrated by anybody, at any time, and any place without restriction, it is clear that they are common at all social levels. These traits make it very difficult to eradicate criminal behavior and necessitate severe management when dealing with them.

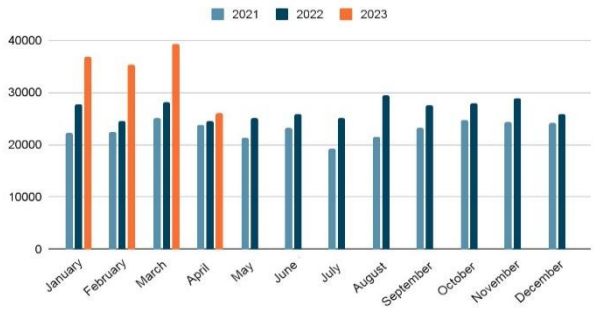

Fig. 1. Crime Data per Month in 2021, 2022, and 2023.

Every criminal case that occurs in Indonesia is specifically examined, tried, and decided through a court as stipulated in Law no. 48 of 2009 concerning Judicial Power [2]. In court, the judge has the duty and responsibility to enforce the law and make decisions in accordance with the rules and provisions contained in the criminal law.

Someone who is proven to have made a mistake and violated the established criminal law will be given sanctions and punishments following the applicable provisions to give a sense of deterrence to the perpetrators. Therefore, the court is an important instrument in eradicating crime and law enforcement in Indonesia. The court is expected to carry out its role effectively and efficiently. However, in reality, the courts in Indonesia are still far from this expectation. Based on a survey conducted by the Indonesian Survey Institute [3], the level of public trust in courts in Indonesia is quite low compared to several other institutions. Furthermore, there is an imbalance between the number of cases and the settlement period in court.

In Indonesia, jurisprudence is used as an addition to the constitution and written legislation to determine the outcome of decisions. Jurisprudence itself is the result of previous court decisions regarding similar legal cases that can be used by judges as a basis for making legal decisions in the cases at hand. If there is a legal vacuum or conditions where there are no established rules for a case, previous decisions made by other judges can become legal instruments to maintain legal certainty. Thus, it can be said

Manuscript received August 22, 2023; revised October 8, 2023; accepted November 14, 2023; published February 24, 2024.

that jurisprudence can encourage increased effectiveness and efficiency of the justice system in Indonesia.

The use of technology to support jurisprudence in order to increase the effectiveness and efficiency of the justice system in Indonesia can be done by creating a system that can predict court decisions based on data from past decisions. Jurisprudence that is properly implemented can also benefit various parties working in the legal field. In addition, the system created can also be useful for education in the field of law and for the general public to provide a legal basis for cases that occur around them. One of the technologies that can be used to solve this problem is through a branch of artificial intelligence, namely Natural Language Processing (NLP). The world of NLP experienced a rapid breakthrough with the introduction of Transformer [4] and Bidirectional Encoder Representations from Transformers (BERT) [5]. BERT is used to solve various kinds of NLP tasks, including text classification and prediction, as well as being used as a baseline in measuring the performance of an NLP model. After its success, the BERT model underwent many modifications to produce other models such as A Robustly Optimized BERT Pretraining Approach (RoBERTa) [6].

This research presents a new model for predicting court decisions in Indonesia from court decision documents. The model is based on the BERT model and its modifications for the Indonesian language. To the authors' knowledge, this is the first model of its kind for legal cases in Indonesia. The model produces predictions in the form of articles of lawsuits imposed on a criminal case in court, based on the existing facts (information) regarding the criminal case.

II. Literature Review

Artificial intelligence is the science that utilizes technology to perform tasks that require human intelligence [7]. Artificial intelligence is related to the process of modeling computers to be able to think and behave like humans. Natural Language Processing (NLP) is a subfield within computer science that focuses on the use of computational techniques with the goal of studying, understanding, and producing human language content [8]. Several fields and areas of need that can be solved through the application of natural language processing, include machine translation, text classification, text summarization, information extraction, conversational agent, question answering system, and speech recognition.

Transformer [4] has become a dominant architecture in the field of natural language processing. The Transformer is said to have surpassed other neural models such as convolutional and recurrent neural networks in terms of performance both in natural language understanding and natural language generation tasks. The architecture of the Transformer follows the encoder-decoder paradigm that is trained from end to end. Without using any recurrent layer, the Transformer model will utilize positional embedding to encode sequences in a sentence.

In Fig. 2, the Transformer uses self-attention and fully connected layers for the encoder (left side) and decoder (right side). Most of the neural sequence transduction models use an encoder-decoder structure. The encoder will map the input sequence from symbol representation $(x_1, \dots, x_n)$ to a continuous representation sequence $z = (z_1, \dots, z_n)$. The decoder will then produce an output sequence $(y_1, \dots, y_n)$ in the form of symbols one element at a time. At each stage, the model is auto-regressive and consumes the previously generated symbols as additional input when generating the next output.

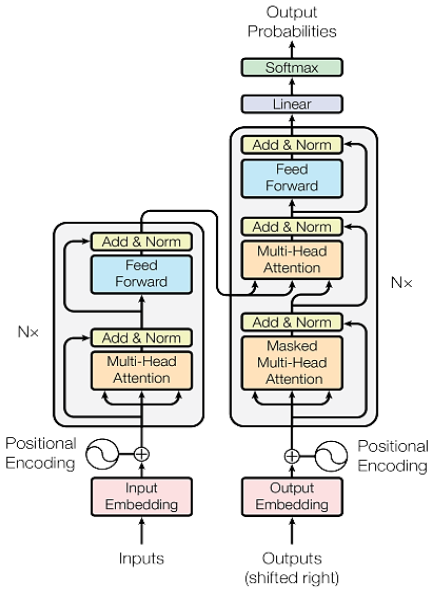

Fig. 2. Transformer model architecture [4].

The attention mechanism used in the Transformer architecture is specifically called "Scaled Dot-Product Attention". The required input consists of queries and keys that have a d_k dimension and values that have a d_v dimension. In practice, the attention function is calculated on a set containing queries directly incorporated into a Q matrix. The keys and values used are also incorporated into K and V matrices. The output matrix can be calculated using the formula:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Devlin *et al.* [5] introduced BERT as a new language model based on the initial implementation of attention on Transformer. BERT is designed to pre-train a bidirectional representation of unlabeled text by conditioning the left and right contexts together in all layers.

BERT is trained using two steps, pre-training and fine-tuning. In the pre-training step, the model is trained based on unlabeled data on two pre-training tasks, namely the Masked Language Model (MLM) and Next Sentence Prediction (NSP). When performing an MLM task, the BERT model will predict the identities of words that have been covered randomly as much as 15% of the total text input. Meanwhile, in NSP, the model will predict whether the second half of the input follows the first half of the corpus or is a random paragraph, with a 50% probability that the second part follows the first part and 50% that the second part is a random paragraph that does not follow the first part. Pre-trained BERT models can be fine-tuned by

making slight modifications to the output produced to perform various other Natural Language Processing tasks, such as text classification, Named Entity Recognition (NER), Question and Answering (QnA), and so on. Fine-tuning of the BERT model can be done in a relatively short time while still giving state-of-the-art performance for various NLP tasks.

Furthermore, Liu *et al.* [6] proposed a new recipe that can be used in the training phase of BERT models called RoBERTa (Robustly Optimized BERT Approach). RoBERTa is said to be able to achieve the same or higher performance than the methods published after BERT. Modifications made in RoBERTa include: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. A new dataset (CC-NEWS) was also used to control for side effects of training set size. The use of this novel dataset also allows decision-making that the use of more data at the pre-training stage can improve the performance of various downstream tasks.

To meet and respond to the needs of NLP in the Indonesian language, several models have been created based on the BERT model and its modifications, such as IndoBERT-IndoNLU [9], IndoBERT-IndoLEM [10], and Indonesian RoBERTa [11].

Research and design of models that can predict court decisions based on court decision documents are still being carried out and developed to this date. Zhong *et al.* [12] create a framework called TopJudge which applies multi-task learning to predict the decision of a legal case in court based on existing facts, where the prediction in question consists of three things, namely law articles, charges, and terms of penalty. Hu *et al.* [13] create a system that can predict the charge of a case by using a description of existing facts. In addition, the focus of that research is to solve some of the challenges faced in predicting charges, such as Few-Shot Charges and Confusing Charges, using a set of discriminatory attributes for each charge in the form of a Yes or No statement for each pair (charge, attribute). Chalkidis *et al.* [14] predict court decisions based on the facts of a legal case, especially in predicting binary violations, multi-label violations, and case importance. Niklaus *et al.* [15] predict the outcome of a court decision of a case, in particular predicting whether the demands filed in a case are granted or rejected. That research also contributes by providing multilingual datasets and benchmarks for legal judgment prediction tasks. The above studies were conducted in different languages, namely English, Mandarin, French, German, and Italian.

In the Indonesian language itself, court decision documents that have been given public access have encouraged various studies to be conducted to utilize the information contained therein. However, previous studies were conducted for different purposes. Nuranti *et al.* [16] predict the categories and length of punishment in first-level court using multi-level learning (CNN+attention). Nuranti *et al.* [17] observed the effectiveness of several machine learning and deep learning methods to recognize 10 legal entities in Indonesian court decision documents. That research is aimed at identifying relevant legal information in court decision documents so that it can be utilized effectively and efficiently. Putri [18] classifies divorce decision documents in court which are divided into two classes, namely talaq divorce and judicial divorce, using the K-Nearest Neighbor algorithm to calculate similarity and classify a document into a class based on the value of cosine similarity.

III. Materials and Methods

As shown in Fig. 3, the steps conducted in this research include dataset collection and parsing, pre-training the BERT model followed by fine-tuning it and doing a model performance comparison and evaluation. The dataset used in this study comes from the Indo-Law dataset [16]. The Indo-Law dataset contains data on court decision documents from the website of the Supreme Court of the Republic of Indonesia (known as Mahkamah Agung Republik Indonesia) that meet the following requirements:

(1) First-level criminal decision (a decision issued by the court of first instance/district court)

(2) Decisions that have permanent legal force (inkracht) obtained from District Courts in West Java, Central Java, East Java, Jakarta, or Yogyakarta

(3) Decision document that has a PDF file

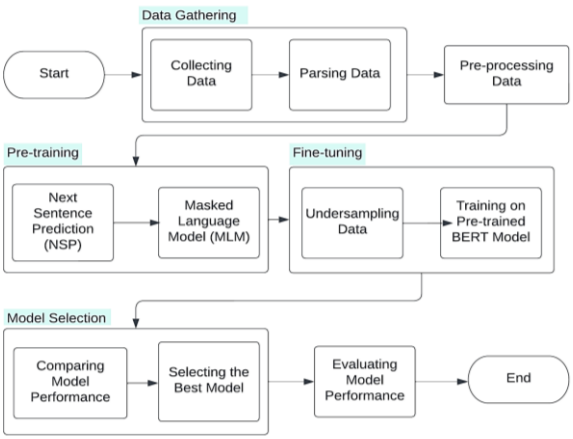

Fig. 3. Our research methodology.

The total number of decision documents contained in the Indo-Law dataset is 22.630 cases. Each decision document is stored in a file in XML format. The contents of each XML file are text from a PDF file of court decision documents from the website of the Indonesian Supreme Court which have gone through a pre-processing process. Each existing XML file is then downloaded and stored on local hardware for use in this study. These pieces of information are contained in the XML file:

(1) Kepala putusan (*document opener*)
(2) Identitas terdakwa (*defendant's identity*)
(3) Riwayat penahanan (*detention history*)
(4) Riwayat perkara (*case history*)
(5) Riwayat tuntutan (*prosecution history*)

(6) Riwayat dakwaan (*indictment history*)
(7) Fakta (*facts*)
(8) Fakta hukum (*legal facts*)
(9) Pertimbangan hukum (*legal considerations*)
(10) Amar putusan (*verdict*)
(11) Penutup (*closing*)

For this research, the fields verdict, facts, and prosecution history are used together with ID, *amar*, classification, and sub-classification from the document's header section. After that, the data is further explored and labeled using the help of regular expression. The resulting label is the expected court decision article for the committed crime.

Furthermore, tokenizing is then done to break the input text into smaller units or tokens. In this study, BertTokenizer from the Transformers library in HuggingFace was used to tokenize fact text. BertTokenizer accepts fact text as input and produces output in the form of input_ids, token_type_ids, and attention_mask. The input_ids contains the mapping between each token and the ID that represents it. Token_type_ids are usually used for tasks such as next sentence prediction and question answering, where there are two paired texts with token_type_ids having a value of 0 for the first text and a value of 1 for the second text. The attention_mask is used as a marker to distinguish the padding token from the original token, where the attention_mask will have a value of 0 for the padding token and 1 for the original token. To make predictions, only input_ids and attention_mask are needed so that token_type_ids can be ignored and will not be used. Fig. 4 shows an example of the tokenizing process using BertTokenizer.

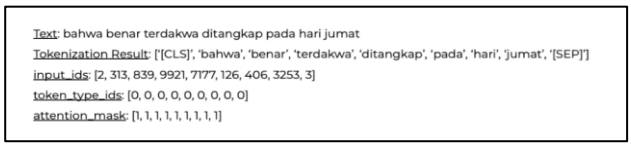

Fig. 4. Tokenizing process using BertTokenizer.

The pre-training stage was carried out by creating a new BERT model based on the IndoBERT-IndoNLU model and the IndoBERT-IndoLEM model. Pre-training is carried out so that the BERT model created can understand the Indonesian language and its context which is specifically used in court decision documents. The data used for pre-training is all data included in the sub-classification of narcotics-and-psychotropics, theft, embezzlement, fraud, abuse, murder, and gambling in the Indo-Law dataset with a total of 14,309 cases, of which only fact part only. The data is divided into 2 parts, namely 85% of the total data as training data and 15% of the total data as data validation.

There are two tasks performed at the pre-training model stage, namely Next Sentence Prediction (NSP) and Masked Language Modeling (MLM). In the NSP task, the BERT model will be trained to predict whether one sentence follows another. The result of the NSP stage for the resulting BERT model is the ability to study long-term relationships between existing sentences.

The fine-tuning stage is then carried out by training a previously pre-trained model to predict the lawsuit articles of a criminal case. The data used for fine-tuning is pre-labeled data, with a total of 14 classes. To overcome the imbalance (imbalance) in the amount of data between classes, undersampling is carried out in several classes to obtain balanced data (balance). Classes with an amount of data greater than 200 data will go through an undersampling process, leaving 200 data selected, while classes with an amount of data less than or equal to 200 data will be used entirely.

The total data used for the fine-tuning stage after undersampling totaled 2,631 cases. The data is then divided into 2 parts, namely 85% of the total data as training data and 15% of the total data as validation data. The training data is used to train the model in predicting lawsuit articles, while the validation data is used to evaluate the model's performance in making predictions for each epoch. In dividing data, the proportion between classes in each section is maintained to remain the same. This can affect the improvement of model performance in making predictions. In fine-tuning, the Adam optimization technique is used to update the weight-weight and learning rate so that the model can minimize loss and improve performance in making predictions.

In this study, fine-tuning was carried out using several different types of pre-trained models as part of an experiment to produce the best model for predicting articles of lawsuits. The models used include the IndoBERT-IndoNLU model which consists of 2 models, namely IndoBERT-base and IndoBERT-large with the names indobert-base-p1 and indobert-large-p1, the IndoBERT-IndoLEM model which consists of 1 model with the name indobert-base-uncased, and the Indonesian RoBERTa model which consists of 1 model with the name indonesian-roberta-base. In addition, the model generated from the pre-training stage based on the IndoBERT-IndoNLU (indobert-base-p1) model is used which is named legal-indobert-indonlu, and the model generated from the pre-training stage based on the IndoBERT-IndoLEM (indobert-base-uncased) which is named legal-indobert-indolem. In addition, implementation and testing of several variations of the BERT model architecture were also carried out, including the BERT base, Hierarchical BERT with Mean Pooling, and Hierarchical BERT with LSTM, as well as testing several hyperparameters such as learning rate and batch size.

The first architecture, BERT Base, is an unmodified BERT architecture as shown in Fig. 5. However, as shown in Fig. 6, in this variant, there are limitations, namely that it can only accept up to 512 tokens of input, so truncation is carried out on fact text that has a length of more than 512 tokens and a tensor is generated from the text containing input_id and attention_mask with dimensions (2, 512).

To tackle the truncation problem, two new variants, Hierarchical BERT with Mean Pooling (shown in Fig. 7) and Hierarchical BERT with LSTM (shown in Fig. 8) are proposed. In this variant, long text facts will be cut or split into several (maximum 4) parts or chunks with each chunk having a maximum length of 512 tokens, so that the

maximum length of text that can be received as input in this variant is 4×512 = 2,048 tokens. For each chunk, a tensor will be generated containing input_id and attention_mask with dimensions (2, 512).

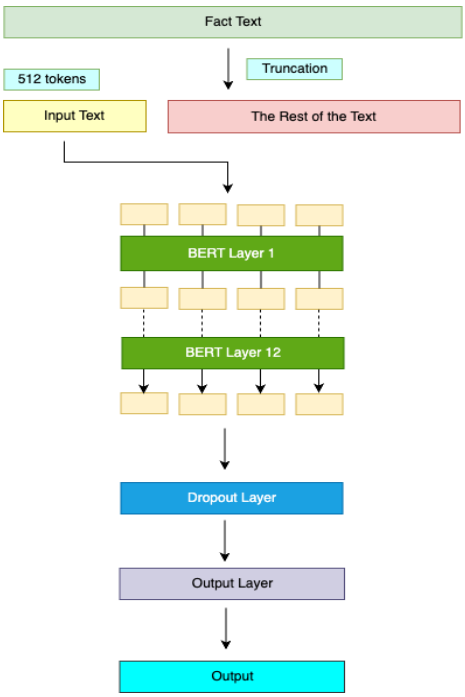

Fig. 5. BERT base architecture.

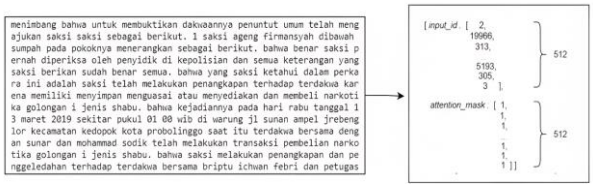

Fig. 6. Text Truncation scheme on Bert base architecture.

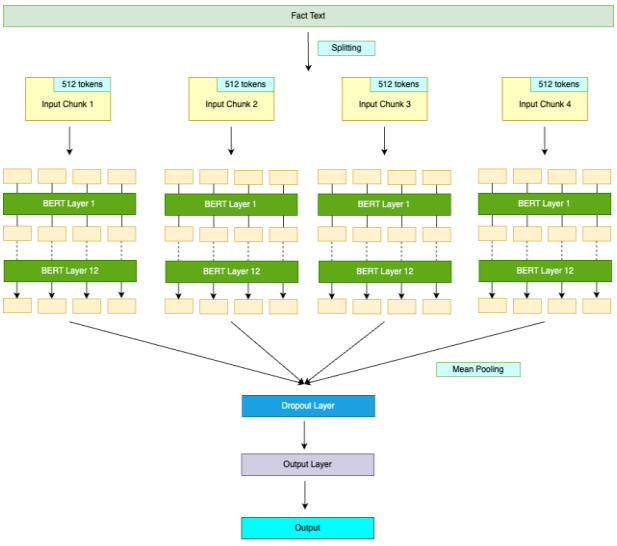

Fig. 7. Hierarchical BERT with mean pooling architecture.

Fig. 9 shows a fact text splitting scheme in the Hierarchical BERT with Mean Pooling and Hierarchical BERT with LSTM variant.

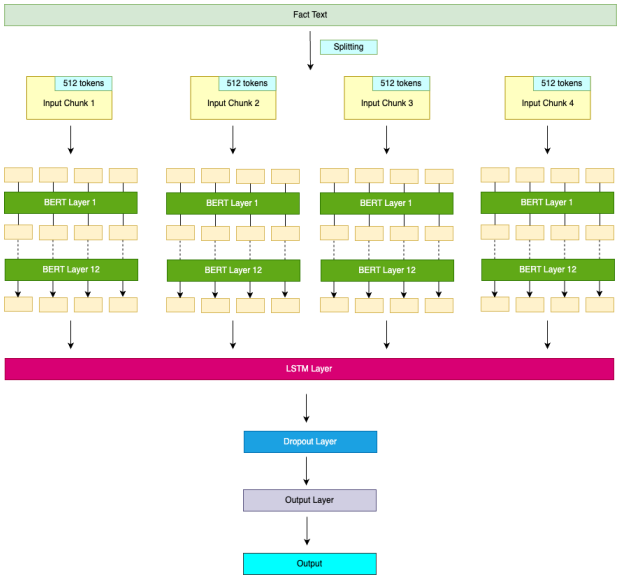

Fig. 8. Hierarchical BERT with LSTM architecture.

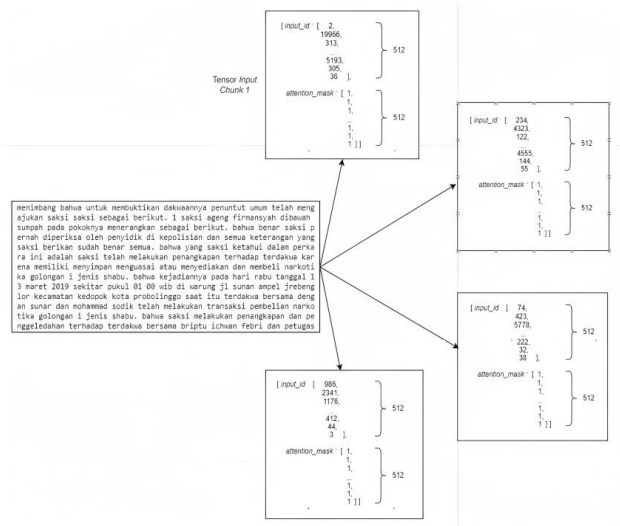

Fig. 9. Improved text truncation scheme on hierarchical BERT + mean pooling and hierarchical BERT + LSTM architecture.

IV. Result and Discussion

The discussion of the experiment results is divided into four main subsections. The first subsection shows the results of the pre-training step, the second subsection shows the results of fine-tuning step, the third subsection shows the comparison and selection of the best model, and the fourth subsection presents the evaluation of the best model.

A. *Pre-training*

Fig. 10 shows the learning curve of the pre-training process for the IndoBERT-IndoNLU model. During the pre-training process, the model succeeded in minimizing the loss value at each epoch for the training data. However, there was a significant increase in the loss value for validation data in the fourth epoch and so on. This shows that the model is experiencing overfitting conditions so it was decided that the pre-training process for the

IndoBERT-IndoNLU would only be carried out until the third epoch to prevent a decrease in model performance in fine-tuning step.

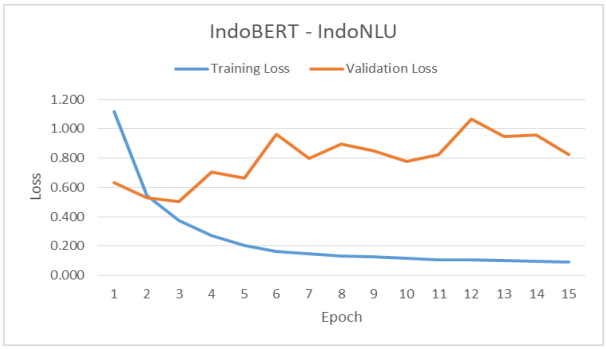

Fig. 10. Learning curve pre-training IndoBERT-IndoNLU model.

Fig. 11 shows the learning curve of the pre-training process for the IndoBERT-IndoLEM model. There is a decrease in the loss value for the training data which indicates that the model is successful in learning from the training data used. However, there was a significant increase in loss values for validation data in the fourth epoch and so on similar to what happened in the pre-training process for the IndoBERT-IndoNLU model. Therefore, it was decided that the pre-training process for the IndoBERT-IndoLEM model would also be carried out until the third epoch to prevent a decrease in model performance in fine-tuning step.

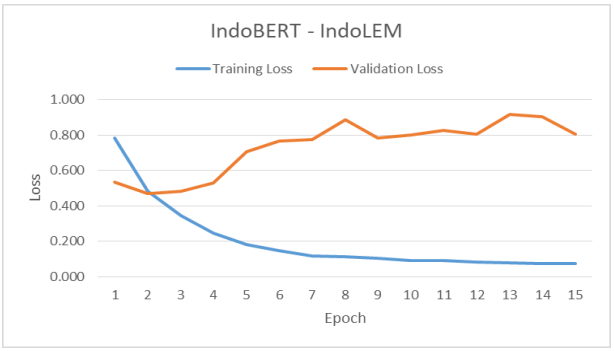

Fig. 11. Learning curve pre-training IndoBERT-IndoLEM model.

B. *Fine-Tuning*

The fine-tuning experiment was carried out with several scenarios using different models, architectures, and hyperparameters. The models to be compared are the IndoBERT-IndoNLU model, the IndoBERT-IndoLEM model, and the Indonesian RoBERTa model, as well as the models resulting from the pre-training process on the IndoBERT-IndoNLU and IndoBERT-IndoLEM models using existing datasets. The architectures to be compared are BERT Base, Hierarchical BERT with Mean Pooling, and Hierarchical BERT with LSTM. The hyperparameters to be compared are learning rate and batch size. The purpose of this hyperparameter comparison is to determine the most optimal hyperparameter in conducting training. The values used as learning rates are 1e–5, 2e–5, 3e–5, and 5e–5, while the values used as batch sizes are 2, 4, and 8. Details of the scenarios are shown in Table I.

TABLE I. EXPERIMENT SCENARIOS

Model Name	Architecture	Learning Rate	Batch Size
indobert-base-p1	BERT Base	2e–5, 5e–5	4, 8
	Hierarchical BERT & Mean Pooling	1e–5, 2e–5	2
	Hierarchical BERT & LSTM	1e–5, 2e–5	2
indobert-base-uncased	BERT Base	2e–5, 5e–5	4, 8
	Hierarchical BERT & Mean Pooling	1e–5, 2e–5	2
	Hierarchical BERT & LSTM	1e–5, 2e–5	2
legal-indobert-indonlu	BERT Base	2e–5, 5e–5	4, 8
	Hierarchical BERT & Mean Pooling	1e–5, 2e–5	2
	Hierarchical BERT & LSTM	1e–5, 2e–5	2
legal-indobert-indolem	BERT Base	2e–5, 5e–5	4, 8
	Hierarchical BERT & Mean Pooling	1e–5, 2e–5	2
	Hierarchical BERT & LSTM	1e–5, 2e–5	2
indobert-large-p1	BERT Base	2e–5, 3e–5	4
indonesian-roberta-base	BERT Base	2e–5, 3e–5	4, 8

1) *Indobert-base-p1*

Table II shows the experiment results for the indobert-base-p1 model in the form of F1-score values (in percent) per epoch from each scenario. The highest F1-score value for each scenario is marked with a yellow highlight.

TABLE II. EXPERIMENT RESULTS FOR THE INDOBERT-BASE-P1 MODEL

Epoch	Base				Hierarchical + Mean Pooling		Hierarchical + LSTM	
	4 batch		8 batch		2 batch		2 batch	
	2e–5	5e–5	2e–5	5e–5	1e–5	2e–5	1e–5	2e–5
1	53.2688	45.0358	53.0891	49.9913	57.2956	59.7466	56.9658	41.5445
2	61.5386	49.9767	62.8728	55.3977	66.5861	66.7222	62.3130	55.2988
3	66.7025	55.0438	70.5373	61.7430	69.5002	70.1731	71.4550	68.8355
4	66.7449	56.0003	64.4544	67.4786	71.1275	**74.0503**	72.6029	68.1335
5	70.3021	65.3557	69.4398	62.7970	**74.2562**	72.8945	72.6198	67.5497
6	70.7464	63.7201	68.6062	69.2141	72.0642	70.4783	72.0321	68.7407
7	72.0351	**71.0322**	70.0164	70.8992	72.1237	71.3209	72.3186	69.4077
8	71.4196	67.6900	**72.0946**	**71.5972**	72.9524	71.4311	73.8433	68.7001
9	71.3412	69.9495	71.0127	70.8204	71.3924	71.7865	**73.9591**	**72.0068**
10	**72.1050**	68.7876	71.8428	70.8479	73.3452	72.5174	72.9269	71.1068

The best performance for the BERT Base architecture is using a batch size of 4 and a learning rate of 2e–5 with an F1-score value of 72.1050%. While the best performance for Hierarchical BERT + Mean Pooling and Hierarchical

BERT + LSTM architectures is using batch size 2 and learning rate 1e−5 with F1-score values of 74.2562% and 73.9591% respectively.

In this model, using the value 2e−5 as the learning rate gets the best performance for the BERT Base architecture, while the value 1e−5 gets the best performance for the Hierarchical BERT + Mean Pooling and Hierarchical BERT + LSTM architectures. In addition, the use of Hierarchical architectures such as Hierarchical BERT + Mean Pooling and Hierarchical BERT + LSTM managed to obtain better performance than BERT Base architecture. This indicates that the two architectures can improve model performance in predicting long texts. Hierarchical BERT + Mean Pooling architecture with batch size 2 and learning rate 1e−5 has the best performance so it was chosen as the scenario to be compared with other models.

2) Indobert-base-uncased

Table III shows the experiment results for the indobert-base-uncased model in the form of F1-score values (in percent) per epoch from each scenario. The highest F1-score value for each scenario is marked with bold.

TABLE III. EXPERIMENT RESULTS FOR THE INDOBERT-BASE-UNCASED MODEL

Epoch	Base				Hierarchical + Mean Pooling		Hierarchical + LSTM	
	4 batch		8 batch		2 batch		2 batch	
	2e–5	5e–5	2e–5	5e–5	1e–5	2e–5	1e–5	2e–5
1	53.7615	51.7136	52.4402	57.9195	62.2050	60.3227	62.3790	55.4562
2	64.9938	58.0896	65.3332	67.9291	69.3527	66.6993	65.8623	68.2056
3	71.9090	69.6821	68.8629	69.3502	73.6002	70.8906	70.3968	75.0894
4	**75.0620**	72.9287	69.7411	69.0389	**75.9358**	**74.5140**	**76.0842**	74.6453
5	71.3687	72.0125	69.9416	72.0060	74.2538	72.5077	74.2222	74.1493
6	71.5885	73.1545	70.2661	70.8490	73.4919	73.3472	72.5693	72.8016
7	72.8695	74.7876	70.4876	**73.7539**	71.7214	71.3205	75.0204	75.4133
8	72.4486	73.4783	**70.9707**	73.3995	74.0669	72.8101	73.0809	76.2230
9	73.1871	74.5513	70.0646	72.9286	74.1193	72.8735	74.0680	76.1759
10	73.0590	**75.0381**	69.2970	72.3721	74.0573	72.5501	74.2631	**76.3610**

The best performance for the BERT Base architecture is using a batch size of 4 and a learning rate of 2e−5 with an F1-score value of 75.0620%. The best performance for the Hierarchical BERT + Mean Pooling architecture is using a batch size of 2 and a learning rate of 1e−5 with an F1-score value of 75.9358%. While the best performance for Hierarchical BERT + LSTM is using batch size 2 and learning rate 2e−5 with an F1-score value of 76.3610%.

In this model, using the value 2e−5 as the learning rate gets the best performance for the BERT Base and Hierarchical BERT + LSTM architectures, while the value 1e−5 gets the best performance for the Hierarchical BERT + Mean Pooling architecture. In addition, the use of Hierarchical architectures such as Hierarchical BERT + Mean Pooling and Hierarchical BERT + LSTM managed to obtain better performance than BERT Base architecture. This indicates that the two architectures can improve model performance in predicting long texts. Hierarchical BERT + LSTM architecture with batch size 2 and learning rate 2e−5 has the best performance so it was chosen as the scenario to be compared with other models.

3) Legal-indobert-indonlu

Table IV shows the experiment results for the legal-indobert-indonlu model in the form of F1-score values (%) per epoch from each scenario. The highest F1-score value for each scenario is marked with bold.

TABLE IV. EXPERIMENT RESULTS FOR THE LEGAL-INDOBERT-INDONLU MODEL

Epoch	Base				Hierarchical + Mean Pooling		Hierarchical + LSTM	
	4 batch		8 batch		2 batch		2 batch	
	2e–5	5e–5	2e–5	5e–5	1e–5	2e–5	1e–5	2e–5
1	49.6629	40.1480	49.3259	43.9703	57.4065	49.3571	56.3582	47.1465
2	57.3105	50.4287	57.6203	53.2555	63.1073	61.1171	61.3845	60.4221
3	64.8794	56.5084	67.6144	55.5530	68.3691	67.4903	69.1186	68.1574
4	68.6137	60.3513	66.5675	56.8189	71.3424	73.1492	71.5440	70.1163
5	70.2996	64.5588	69.7129	63.9130	**73.6766**	72.5883	73.5695	73.9762
6	70.7573	66.7563	71.7062	69.4975	71.7094	71.2362	71.9557	70.2885
7	70.1354	67.8733	**72.5803**	68.7521	70.2272	74.6981	72.5530	74.6459
8	71.9905	**71.2647**	71.8287	69.9831	72.7532	71.8730	72.6319	74.9435
9	71.3931	68.7720	71.4769	**70.8056**	72.4019	74.1772	72.8500	73.5574
10	**72.3833**	69.2530	72.0718	70.7287	72.8092	**73.5197**	73.8194	**76.4948**

The best performance for the BERT Base architecture is using a batch size of 8 and a learning rate of 2e−5 with an F1-score value of 72.5803%. While the best performance for Hierarchical BERT + Mean Pooling and Hierarchical BERT + LSTM architectures is using batch size 2 and learning rate 2e−5 with F1-score values of 74.6981% and 76.4948%, respectively.

In this model, using the value 2e−5 as the learning rate managed to get the best performance compared to other values. In addition, the use of Hierarchical architectures such as Hierarchical BERT + Mean Pooling and Hierarchical BERT + LSTM managed to obtain better performance than BERT Base architecture. This indicates that the two architectures can improve model performance

in predicting long texts. Bert + LSTM Hierarchical Architecture with batch size 2 and learning rate 2e−5 has the best performance so it was chosen as the scenario to be compared with other models.

4) Legal-indobert-indolem

Table V shows the experiment results for the legal-indobert-indolem model in the form of F1-score values (in percent) per epoch from each scenario. The highest F1-score value for each scenario is marked with a bold.

The best performance for the BERT Base architecture is using a batch size of 4 and a learning rate of 2e−5 with an F1-score of 74.0767%. The best performance for the Hierarchical BERT + Mean Pooling architecture is using a batch size of 2 and a learning rate of 1e−5 with an F1-score value of 79.8888%. While the best performance for Hierarchical BERT + LSTM is using batch size 2 and learning rate 2e−5 with an F1-score value of 77.8415%.

TABLE V. EXPERIMENT RESULTS FOR THE LEGAL-INDOBERT-INDOLEM MODEL

Epoch	Base				Hierarchical + Mean Pooling		Hierarchical + LSTM	
	4 batch		8 batch		2 batch		2 batch	
	2e–5	5e–5	2e–5	5e–5	1e–5	2e–5	1e–5	2e–5
1	47.4220	55.1256	46.4280	38.6872	69.0083	61.9760	61.6047	58.3340
2	63.9711	52.4503	61.1992	51.2908	69.9582	68.4664	69.7253	68.9382
3	71.2459	59.9782	67.6551	56.9034	73.7658	72.9344	71.1592	74.6344
4	71.9196	68.7601	**72.2298**	53.0137	76.2029	77.1701	76.6894	76.6971
5	69.8122	73.2022	70.6068	60.8200	78.2598	74.3330	73.9290	76.0817
6	71.1799	71.0772	70.4261	62.3644	77.7260	**77.2894**	**76.7115**	**77.8415**
7	73.7960	**73.2864**	71.0913	70.1407	77.7286	74.9304	73.1349	75.5916
8	72.9009	70.9633	70.6563	70.1908	76.3413	74.0284	75.0779	74.9408
9	73.5834	72.5663	69.9943	69.3489	**79.8888**	76.6830	74.7799	77.1945
10	**74.0767**	72.2058	70.3994	**70.9608**	78.8397	75.3882	75.7570	77.2513

In this model, using the value 2e−5 as the learning rate gets the best performance for the BERT Base and Hierarchical BERT + LSTM architectures, while the value 1e−5 gets the best performance for the Hierarchical BERT + Mean Pooling architecture. In addition, the use of Hierarchical architectures such as Hierarchical BERT + Mean Pooling and Hierarchical BERT + LSTM managed to obtain better performance than BERT Base architecture. This indicates that the two architectures can improve model performance in predicting long texts. Hierarchical BERT + Mean Pooling architecture with batch size 2 and learning rate 1e−5 has the best performance so it was chosen as the scenario to be compared with other models.

5) Indobert-large-p1

Table VI shows the experiment results for the indobert-large-p1 model in the form of F1-score values (in percent) per epoch from each scenario. The highest F1-score value for each scenario is marked with a yellow highlight.

TABLE VI. EXPERIMENT RESULTS FOR THE INDOBERT-LARGE-P1 MODEL

Epoch	Base	
	4 batch	
	2e–5	3e–5
1	50.0006	47.9028
2	67.3683	50.1059
3	69.0759	59.8342
4	70.9403	66.7315
5	70.7528	72.8977
6	69.7958	68.6740
7	**73.3959**	**74.1498**
8	72.3829	70.2920
9	71.8268	72.4182
10	71.9443	71.5052

The best performance obtained in this model is using a batch size of 4 and a learning rate of 3e−5 with an F1-score of 74.1498%. This scenario will be selected to be compared with other models.

6) Indonesian-roberta-base

Table VII shows the experiment results for the indonesian-roberta-base model in the form of F1-score values (in percent) per epoch from each scenario. The highest F1-score value for each scenario is marked with bold.

TABLE VII. EXPERIMENT RESULTS FOR THE INDONESIAN-ROBERTA-BASE MODEL

Epoch	Base			
	4 batch		8 batch	
	2e–5	3e–5	2e–5	3e–5
1	55.7948	49.8668	42.6865	46.6933
2	60.8747	56.9512	54.6242	58.5029
3	69.1562	68.8815	58.2525	59.5363
4	67.6978	68.7743	62.0725	58.0582
5	69.1759	67.2392	61.1205	63.4305
6	70.2345	**70.3197**	66.3353	68.7378
7	69.5982	69.1189	68.5112	**71.7251**
8	**71.2574**	69.6780	68.9540	69.6924
9	70.4623	70.2316	**69.5792**	70.1131
10	70.5726	70.3498	69.4102	70.5418

The best performance obtained in this model is using a batch size of 8 and a learning rate of 3e−5 with an F1-score value of 71.7251%. This scenario will be selected to be compared with other models.

C. Comparison and Selection of the Best Model

In this study, Support Vector Machine (SVM) + TF-IDF was used as a baseline to compare the performance of BERT models and their modifications with traditional machine learning models. In implementing the SVM model, experiments were carried out using 3 types of kernels, namely linear, polynomial, and RBF. The results of the SVM + TF-IDF model created are shown in Table VIII.

TABLE VIII EXPERIMENT RESULTS FOR THE SVM + TF-IDF MODEL

Kernel	F1-Score
Linear	**58.4810%**
Polynomial	34.1772%
RBF	56.2025%

The best performance in the SVM + TF-IDF model is obtained by using a linear kernel with an F1-score value of 58.4810%. This scenario was chosen as the baseline to measure the performance of the BERT models and their modifications resulting from the fine-tuning experiments that have been carried out. In this section, a comparison is made between each of the best scenarios of each model and the baseline to select the best model that will be used as the main model in predicting the articles of lawsuits from a criminal case.

Based on Table IX, the SVM + TF-IDF model with a linear kernel as a baseline obtained the worst performance compared to other models with an F1-score value of 58.4810%, this shows that the Transformer BERT model succeeded in outperforming the traditional machine learning model in performing predictions of legal articles. The best performance was achieved by the legal-indobert-indolem model with the Hierarchical BERT + Mean Pooling architecture, learning rate 1e−5, and batch size 2, where the F1-score value obtained was 79.8888%. This model managed to beat the indobert-base-p1 and indobert-base-uncased models which were produced without going through the pre-training step with a difference in F1-scores of 5.6326% and 3.5278%. This model also obtained a higher F1-score value of 3.394% than the legal-indobert-indonlu model which was also produced from the pre-training step. In addition, the indobert-large-p1 and indonesian-roberta-base models were also outperformed with significant differences in performance, in the amount of 5.739% and 8.1637%. With these results, the legal-indobert-indolem model with the Hierarchical BERT + Mean Pooling architecture, learning rate 1e−5, and batch size 2, will be chosen as the main model in predicting the lawsuit articles of a criminal case.

TABLE IX. BEST SCENARIO FOR EACH MODEL AND BASELINE

Model Name	Architecture	Learning Rate	Batch Size	Kernel	F1-score
SVM + TF-IDF	-	-	-	*Linear*	58.4810%
indobert-base-p1	*Hierarchical* BERT + *Mean Pooling*	1e−5	2	-	74.2562%
indobert-base-uncased	*Hierarchical* BERT + LSTM	2e−5	2	-	76.3610%
legal-indobert-indonlu	*Hierarchical* BERT + LSTM	2e−5	2	-	76.4948%
legal-indobert-indolem	*Hierarchical* BERT + *Mean Pooling*	1e−5	2	-	**79.8888%**
indobert-large-p1	BERT *Base*	3e−5	4	-	74.1498%
indonesian-roberta-base	BERT *Base*	3e−5	8	-	71.7251%

D. *Evaluation of the Best Model*

In this section, the performance of the legal-indobert-indolem model with the Hierarchical BERT + Mean Pooling architecture, learning rate 1e−5, and batch size 2 will be discussed in more detail, as the chosen model for predicting lawsuit articles.

The model was trained using 85% of all fine-tuning data for 10 epochs. At each epoch, an evaluation of the predictions of the model is carried out using 15% of the entire fine-tuning data for validation. Loss values from the training and validation processes in each epoch will be compared with the F1-score values obtained.

Fig. 12 shows that there is a decrease in the loss value at each epoch for the training process, which means that the model is successful in learning from the training data used. On the other hand, there was an increase in the loss value for the validation process in the fifth to eighth epoch, and it decreased again in the ninth epoch. This was followed by the F1-score value which increased in the first to fifth epoch, then decreased until the eighth epoch, and increased until it reached the highest F1-score value in the ninth epoch.

Fig. 13 shows the confusion matrix of the predictions made by the selected model. Overall, the model succeeded in predicting lawsuit articles quite well on validation data. In predicting several articles that fall into the same sub-classification (such as Article 338 and Article 340 which are included in the sub-classification of murder), there are several prediction errors considering that the fact of these articles tends to be quite similar, but the number is not too much. On the other hand, in predicting articles belonging to different sub-classifications, such as Article 351 concerning the sub-classification of abuse and Article 303 concerning the sub-classification of gambling, the model has succeeded in predicting them accurately.

Fig. 12. Learning curve of the chosen model.

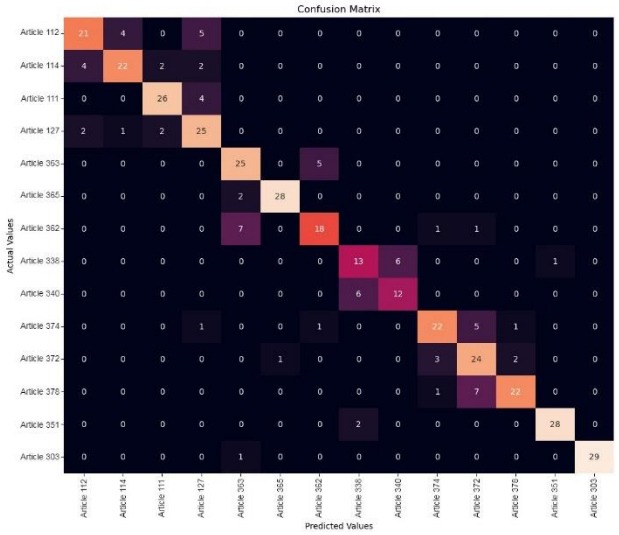

Fig. 13. Confusion matrix of the prediction results.

Table X shows the evaluation of the selected model using precision, recall, and F1-score for each class and on average (weighted average). In each class, the model managed to obtain good performance for precision, recall, and F1-score with values of more than 60%. On average (weighted average), the model managed to obtain a precision value of 80.5930%, a recall of 79.7468%, and an F1-score of 79.8888%. This indicates that the model has good performance in making predictions.

TABLE X. MODEL EVALUATION RESULTS WITH EVALUATION METRICS

Class	Precision (%)	Recall (%)	F1-score (%)
Article 112	77.7778	70.0000	73.6842
Article 114	81.4815	73.3333	77.1930
Article 111	86.6667	86.6667	86.6667
Article 127	67.5676	83.3333	74.6269
Article 363	71.4286	83.3333	76.9231
Article 365	96.5517	93.3333	94.9153
Article 362	75.0000	66.6667	70.5882
Article 338	61.9048	65.0000	63.4146
Article 340	66.6667	66.6667	66.6667
Article 374	81.4815	73.3333	77.1930
Article 372	64.8649	80.0000	71.6418
Article 378	88.0000	73.3333	80.0000
Article 351	96.5517	93.3333	94.9153
Article 303	100.0000	96.6667	98.3051
Weighted Average	80.5930	79.7468	79.8888

V. CONCLUSION

From this research, it can be concluded that a system to predict criminal court decisions in Indonesia has been successfully designed and built. A total of 6 BERT and RoBERTa models: *indobert-base-p1*, *indobert-base-uncased*, *legal-indobert-indonlu*, *legal-indobert-indolem*, *indobert-large-p1*, and *indonesian-roberta-base* have been used on top of 3 proposed architectures: BERT Base, Hierarchical BERT + Mean Pooling, and Hierarchical BERT + LSTM. A combination of the model *legal-indobert-indonlu* and the architecture Hierarchical BERT + Mean Pooling has achieved the highest F1-score of 79.8888%. Those combinations of models and architectures are also compared with SVM+TF-IDF as the baseline using linear, polynomial, and RBF kernels. However, the baseline model only obtained the best F1-score of 58.4810% on a linear kernel. Therefore, it can also be concluded that machine learning methods had relatively low performance compared to deep learning models using BERT modifications. A subjective evaluation conducted using interviews and questionaries has also proven that the proposed model can be used to support Indonesia's jurisprudence in deciding the decision of a criminal case. However, a bigger dataset from the Republic of Indonesia's criminal court decision documents most likely will benefit future research since in this research, several sub-classifications only have 1 document in the dataset. A machine with higher and more robust resources will also allow the usage of higher parameter values, including learning rate and batch size.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

S.L. conducted the research, analyzed the data, and wrote the paper. S.F. conducted the research, analyzed the data, and wrote the paper. D.S. designed the research framework, reviewed the paper, and all authors had approved the final version.

REFERENCES

[1] Crime data. [Online]. Available: https://pusiknas.polri.go.id/data_kejahatan (in Indonesian)
[2] Law of the Republic of Indonesia No. 48 of 2009 concerning Judicial Power.
[3] LSI Survey Release March 1, 2023. [Online]. Available: https://www.lsi.or.id/post/rilis-survei-lsi-01-maret-2023 (in Indonesian)
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint, arXiv:1810.04805v2, 2018.
[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint, arXiv:1907.11692, 2019.
[7] H. Surden, "Artificial intelligence and law: An overview," *Georgia State University Law Review*, vol. 35, pp. 19–22, 2019.
[8] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015. doi: 10.1126/science.aaa8685
[9] B. Wilie, K. Vincentio, G. I. Winata, S. Cahyawijaya, X. Li, Z. Y. Lim, S. Soleman, R. Mahendra, P. Fung, S. Bahar, and A. Purwarianti, "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," arXiv preprint, arXiv:2009.05387, 2020.
[10] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," arXivpreprint, arXiv:2011.00677, 2020.
[11] Flax Community. Indonesian RoBERTa base. [Online]. Available: HuggingFace: https://huggingface.co/flax-community/indonesian-roberta-base
[12] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal judgment prediction via topological learning," in *Proc. the 2018 Conference onEmpirical Methods in Natural Language Processing*, 2018, pp. 3540–3549.
[13] Z. Hu, X. Li, C. Tu, Z. Liu, and M. Sun, "Few-shot charge prediction with discriminative legal attributes," in *Proc. the 27th International*

Conference on Computational Linguistics, August, 2018, pp. 487–498.
[14] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural legal judgment prediction in English," arXiv preprint, arXiv:1906.02059, 2019.
[15] J. Niklaus, I. Chalkidis, and M. Stürmer, "Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark," arXiv preprint, arXiv:2110.00806, 2021.
[16] E. Q. Nuranti, E. Yulianti, and H. S. Husin, "Predicting the category and the length of punishment in Indonesian courts based on previous court decision documents," *Computers*, vol. 11, no. 6, p. 88, 2022.
[17] E. Q. Nuranti and E. Yulianti, "Legal entity recognition in Indonesian court decision documents using Bi-LSTM and CRF approaches," in *Proc. 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE, 2020, pp. 429–434.
[18] F. L. Putri, "Classification of divorce decision documents using the K-Nearest neighbor method," Doctoral dissertation, Universitas Muhammadiyah Malang, 2021. (in Indonesian)